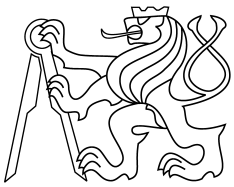
CENTER FOR
MACHINE PERCEPTION



CZECH TECHNICAL
UNIVERSITY IN PRAGUE

PhD THESIS

ISSN 1213-2365

Robust Multiview Reconstruction

Daniel Martinec

martid1@cmp.felk.cvut.cz

CTU–CMP–2008–01

July 2, 2008

Available at
ftp://cmp.felk.cvut.cz/pub/cmp/articles/martinec/Martinec-thesis.pdf

Thesis Advisor: Ing. Tomáš Pajdla, PhD.

Research Reports of CMP, Czech Technical University in Prague, No. 1, 2008

Published by

Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University
Technická 2, 166 27 Prague 6, Czech Republic
fax +420 2 2435 7385, phone +420 2 2435 7637, www: http://cmp.felk.cvut.cz

Robust Multiview Reconstruction

A Dissertation Presented to the Faculty of the Electrical Engineering of the Czech Technical University in Prague in Partial Fulfillment of the Requirements for the Ph.D. Degree in Study Programme No. P 2612 - Electrotechnics and Informatics, branch No. 3902V035 - Artificial Intelligence and Biocybernetics, by

Daniel Martinec

July 2, 2008

Thesis Advisor

Ing. Tomáš Pajdla, PhD.

Center for Machine Perception
Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague
Karlovo náměstí 13, 121 35 Prague 2, Czech Republic
fax: +420 224 357 385, phone: +420 224 357 465
http://cmp.felk.cvut.cz

Abstract

Reconstructing a 3-dimensional (*3D*) model of a scene from a set of 2D images is a fundamental problem in computer vision with many applications. The problem can be decomposed into three steps. First, some correspondences between pairs of images are found and 3D geometries of the image pairs are estimated. Secondly, the two-view geometries are fused into a consistent reconstruction of all views. Thirdly, having a complete camera calibration, a consistent dense model of the scene surfaces can be reconstructed using all images.

While the two-view camera calibration is a well studied problem, the multiview camera calibration remains a challenging task. It is also the most crucial step in the scene reconstruction as the quality of the resulting dense 3D model is fundamentally limited by precision of the multiview camera calibration.

This thesis studies mainly the problem of multiview camera calibration. The largest difficulty of the problem is *sparsity* of the data which happens when the images are only sparsely captured (so-called wide baseline stereo, *WBS*). Then, the scene contains many occlusions, i.e. many points are seen in a few images only. The second difficulty of the problem is handling of incorrect correspondences (*mismatches*), thanks to which also *non-existent pair-wise geometries* can be found. Every such geometry must be detected and removed to obtain a correct reconstruction.

The main contribution of the thesis is a technique for multiview camera calibration by *gluing partial reconstructions*. This technique was used for uncalibrated cameras to obtain a projective reconstruction as well as for partially calibrated cameras to obtain a metric reconstruction. The technique works in practical situations, i.e. the perspective camera, many (99.9%) occlusions in scene and a not entirely exact correspondence algorithm.

The importance of such technique lies in that it offers united and elegant way of processing correspondences from WBS and sequences. The presented methods exploit all data known about the scene, namely in the same way and at once. The core of the methods is a linear algorithm which provides a very good reconstruction already before non-linear refinement using bundle adjustment.

The developed methods embrace projective factorization for points and lines, gluing of projective pairwise reconstructions, merging metric panoramas and gluing pairwise metric reconstructions. Some of the methods are applicable for both affine and perspective camera models. The methods are suited for *large-scale* reconstructions (thousands of images). The accuracy, applicability and speed of the methods is demonstrated on difficult wide baseline image sets whose metric dense reconstructions are shown.

The presented techniques were used in a complete, robust automatic multiview reconstruction pipeline from images to a 3D model.

Acknowledgements

My grateful thanks belong to Tomáš Pajdla for bringing my attention to 3D reconstruction in my diploma thesis and for his excellent guidance and critical comments during all those years. My PhD study was first advised by Mirko Navara and co-advised by Tomáš Pajdla. Václav Hlaváč provided an ideal environment for focusing fully on research at Center for Machine Perception (CMP) in Prague. I thank my colleagues at CMP for creating a friendly atmosphere and for their openness in cooperation. I acknowledge especially the following people for fruitful discussions: Radim Šára, Jiří (George) Matas, Ondřej Chum, Michal Perďoch, Štěpán Obdržálek, Jana Kostková, Jan Čech, Vladimír Smutný, Pavel Krsek, and Martin Bujňák.

Contents

Keywords: computer vision, 3D reconstruction, multiple view reconstruction, auto-calibration, epipolar geometry, mismatch identification, dense stereo, omnidirectional camera, line reconstruction

1 Introduction

This chapter introduces the PhD thesis by explaining its motivation (section 1.1), contributions (section 1.2), and structure (section 1.3).

1.1 Motivation

Reconstruction of a 3D model of a scene from a set of 2D images capturing the scene from different points of view belongs to computer vision problems studied for decades [31]. Such 3D model describes the structure (shape) of the scene and the configuration (motion) of the cameras using which the images were obtained. It is a difficult problem, which is known to be NP-hard [79] when missing data is allowed, i.e. when some point is not visible in some image. Fortunately, sub-optimal solutions are possible. Their study is the objective of this thesis.

There are two qualitatively different approaches to obtaining images of a scene which also determine the nature and the difficulty of the reconstruction problem. In the first approach, a dense sequence of images is obtained while in the latter, images are sparsely captured from possibly very different view points.

In a sequence, consecutive images differ only slightly and there are small distances between the consecutive points of view. The line segment joining two camera centers is called *baseline*. As opposed to narrow baselines in a sequence, where finding correspondences is quite easy, the correspondence problem in *wide baseline stereo (WBS)* becomes very difficult. The two approaches differ not only in difficulty of the correspondence problem but also in the nature of the subsequent reconstruction. From now on it will be supposed that the correspondence problem is solved at least so that some large subset of the correspondences between image pairs is correct.

There exist multi-linear constraints between two, three, and four images. These constraints define tensors which exactly express the two-, three-, and four-view geometry, respectively [31]. However there is no single multi-linear constraint expressing the consistency among more than four images, and thus no such single tensor. Nevertheless, it is possible to express multiview geometry using a matrix equation capturing projections of all points into all images.

Provided that the cameras capture the scene from relatively large distance, a simplified, so-called *affine* or *orthographic*, camera model can be used. Under the affine projection, *image measurements*, i.e. projections of all points into all images, can be directly factorized into (projective) *structure* (point locations) and *motion* (camera locations) [119]. But this is not possible for the (full) *perspective* camera model which brings so-called *(projective) depths* into the basic projection equation. Consequently, each image measurement in the image measurement matrix has an unknown depth that has to be computed before the factorization.

It is possible to estimate the projective depths using the multi-linear constraints.[1] But because the multi-linear constraints exist only for up to four images, they have to be combined from different image subsets. This is equivalent to gluing partial reconstructions from few images together. In an image sequence, the way of joining the partial reconstructions is straightforward: the actual reconstruction is improved using the consecutive image. On the other hand, in wide

[1] Other ways of estimating the projective depths are mentioned in section 3.1.

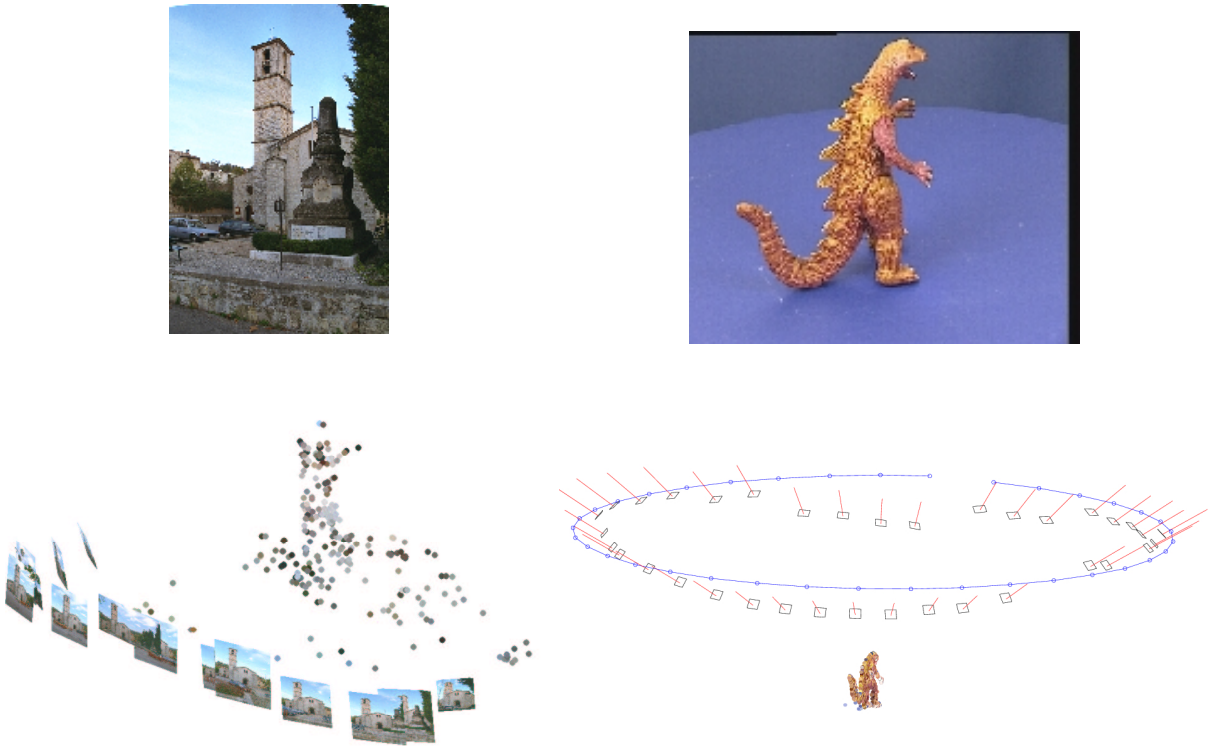


Figure 1.1: *The same approach for* wide baseline multiview stereo (left) and a sequence (right). Shown on the Valbonne image set [63] and the Dinosaur sequence [65] (section 6.3).

baseline case, there are many ways of joining the partial reconstructions and it becomes non-trivial to find a good one among them.

There is a method [43] solving the reconstruction for the affine camera model even when there are *occlusions* in scene, i.e. when not all observed 3D points are visible in all images. However, it turned out that the method is applicable for at most a few tens of images (see section 6.3.1). On the other hand, for the perspective camera model, there exists a method [113] usable only in absence of occlusions, which makes the method almost unusable in practise. Other methods rely on some particular data, e.g., [127].

As for the perspective camera model, several methods for reconstruction from sequences appeared, e.g. [21, 7], but there existed no method for reconstruction of WBS scenes which would exploit all known correspondences in a unified manner. (Method of Guilbert [27] uses only affine camera model which is not suitable for wide baseline setup, see section 6.3.4.)

This thesis brings new methods for 3D reconstruction which are particularly suited for wide baseline multiview stereo. These methods can be also used for sequences because the used approach is general w.r.t. configuration of occlusions in scene as well as camera positions. This is demonstrated by figure 1.1. Both affine and perspective camera models are considered.

Line Features

So far only point correspondences have been discussed. Besides these it is possible to search for line features in images and to establish correspondences among them. Line correspondences are attractive for a number of reasons. There are many lines in man-made environments. Lines can be detected more precisely than points in some situations. Lines are less affected by occlusions as projections of different parts of the same line can be used to reconstruct it. An example of such line correspondences is shown in figure 1.2.

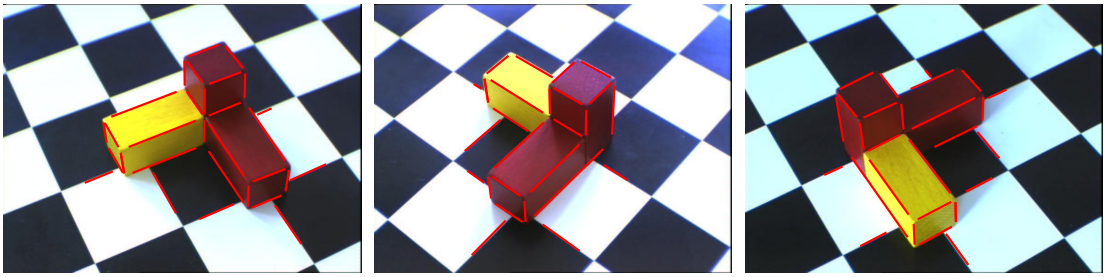


Figure 1.2: *Line correspondences.* Line correspondences can often be established even though line end-points are occluded.

Existing methods for line reconstruction exploiting all known data at once were usable for affine camera model only like [96], or relied on point correspondences like [125]. Further novelty of this thesis are methods for reconstruction of lines for the perspective camera model. An extension towards occlusions and mismatches is presented.

1.2 Contributions of the Thesis

The main contribution of this work is a technique for reconstructing consistent metric geometry of cameras and sparse feature points from many unorganized images by *gluing partial reconstructions*. This technique was used for uncalibrated cameras to obtain a projective reconstruction [65] (section 6.3) as well as for partially calibrated cameras to obtain a metric reconstruction [66, 67] (sections 6.7 and 6.8). The technique works in practical situations, i.e. the perspective camera, many (99.9%) occlusions in scene (figure 6.32, *p*109) and a not entirely exact correspondence algorithm.

The importance of such technique lies in that it offers united and elegant way of processing correspondences from WBS and sequences (figure 1.1). The presented methods exploit all data known about the scene, namely in the same way and at once. The core of the methods is a linear algorithm which provides very good reconstructions already before non-linear refinement using bundle adjustment.

The most important contributions of the thesis:

1. A method for projective gluing of partial reconstructions with (published in [65], section 6.3) and without depth consistency (section 6.5). Jacobs' method [43] was reformulated in the original instead of complementary subspaces.

2. A method for metric gluing of partial reconstructions. Published in [66, 67] (sections 6.7 and 6.8). Very high robustness is achieved w.r.t. occlusions (99%), mismatches and even non-existent epipolar geometries.

3. A panorama detection (section 5.5), building and merging algorithm (section 6.6). Results published in [59].

4. A heuristics for speeding up matching image pairs in large datasets (section 5.1).

5. A relative pose estimation algorithm for unknown focal length ratio (section 4.2).

6. A linear method for triangulation using projective depths (section 4.5).

7. Exploiting camera calibration in DEGENSAC [16] (section 4.3).

8. An extension of DEGENSAC [16] for three views (section 4.4).

9. An extension of the factorization of point correspondences with occlusions from affine to perspective cameras. Published in [62] (section 6.1).

10. A generalization of our reconstruction technique for so called *omni-directional* cameras [74], i.e. perspective cameras which cannot be approximated by the affine model. These cameras have typically large angle of view. Published in [73].

11. A factorization technique for line correspondences from perspective cameras. Published in [64] (chapter 7).

12. Fusion of our reconstruction technique with the correspondence estimator for wide baseline stereo [69]. Published in [63].

13. Exploiting the fact that the cameras are *directional*, which means that rays casted from space into the camera are *oriented*. This information is used for outlier detection in [73].

14. An extension of Kahl's L_∞-norm estimation of multiview geometry [44] w.r.t. mismatches (section 4.6).

15. Detection of wrong correspondences using voting by trifocal tensors. Published in [61] (section 6.2).

History of the developed methods for multiview reconstruction is summarized in table 6.1, *p*42, which contains also examples of scenes illustrating abilities of individual methods. For more examples of the reconstructed scenes see demo pages [1, 2, 3].

1.3 Structure of the Thesis

The multi-view reconstruction problem is formulated in chapter 2 for points and lines. The state of the art is given in chapter 3. Section 3.5 describes the CMP automatic 2D to 3D data pipeline, part of which are the presented reconstruction methods. The output of the pipeline (a 3D dense model) serves to demonstrate the accuracy of the presented methods across the thesis. Chapter 4 describes some achievements in geometry of few views like triangulation, RANSAC [31] and usage of image triplets. Chapter 5 describes our reconstruction pipeline for multiview reconstruction, namely techniques like matching the image pairs, joining matches into tracks, bundle adjustment and detection of image pairs with very short baseline. Chapter 6 describes the developed methods for multi-view reconstruction from point correspondences. Their history can be seen in table 6.1, *p*42 while method overview is given at the beginning of the chapter. Chapter 7 deals with reconstruction from lines. Chapter 8 concludes the thesis. Chapter A proposes an attempt to find a best rank-one approximation to a matrix with the missing elements.

In this work, when referencing some part of it, we usually cite also the publication in which the result was first published. If the two works differ, the thesis is correct.

2 Problem Formulation

At first, reconstruction from point correspondences will be formulated. Our formulation of reconstruction from line correspondences will follow.

2.1 Points

Suppose a set of n 3D points and that some of them are visible in m perspective images. There may be outliers, i.e. mismatches, in image measurements. The goal is to reject outliers and to recover 3D structure (point locations) and motion (camera locations) from the remaining image measurements.

Let $\mathbf{X}_p$ be the unknown homogeneous coordinate vectors of the 3D points, $\mathtt{P}^i$ the unknown 3×4 projection matrices, and $\mathbf{x}_p^i$ the measured homogeneous coordinate vectors of the image points, where $i = 1, \ldots, m$ labels images and $p = 1, \ldots, n$ labels points. Due to occlusions, $\mathbf{x}_p^i$ are unknown for some i and p.

The basic image projection equation says that $\mathbf{x}_p^i$ are the projections of $\mathbf{X}_p$ up to unknown scale factors λ_p^i, which will be called (*projective*) *depths*:

$$\lambda_p^i \mathbf{x}_p^i = \mathtt{P}^i \mathbf{X}_p. \tag{2.1}$$

The complete set of image projections can be gathered into a matrix equation:

$$\underbrace{\begin{bmatrix} \lambda_1^1\mathbf{x}_1^1 & \lambda_2^1\mathbf{x}_2^1 & \ldots & \lambda_n^1\mathbf{x}_n^1 \\ \times & \lambda_2^2\mathbf{x}_2^2 & \ldots & \times \\ \vdots & & \ddots & \vdots \\ \lambda_1^m\mathbf{x}_1^m & \times & \ldots & \lambda_n^m\mathbf{x}_n^m \end{bmatrix}}_{\mathtt{R}} = \underbrace{\begin{bmatrix} \mathtt{P}^1 \\ \vdots \\ \mathtt{P}^m \end{bmatrix}}_{\substack{\mathtt{P} \\ 3m \times 4}} \underbrace{[\mathbf{X}_1 \ldots \mathbf{X}_n]}_{\substack{\mathtt{X} \\ 4 \times n}} \tag{2.2}$$

where marks $\times$ stand for unknown elements which could not be measured due to occlusions, $\mathtt{X}$ and $\mathtt{P}$ stand for structure and motion, respectively. The $3m \times n$ matrix $[\mathbf{x}_p^i]_{i=1\ldots m, p=1\ldots n}$ will be called the *measurement matrix* (MM) whereas $\mathtt{R}$ will be called the *partially rescaled measurement matrix* (PRMM) because $\mathtt{R}$ will be used even with some unknown depths. Both MM and PRMM often have some missing elements and mismatches (outliers).

Several methods will be described that either (i) estimate a projective reconstruction and upgrade it to a metric one (sections 6.1–6.5) or (ii) estimate a metric reconstruction directly (sections 6.6–6.8). In this work, a *metric reconstruction* denotes a reconstruction up to *similarity*, i.e. up to an overall rotation, translation and scale.

2.2 Lines

Suppose a set of n 3D lines visible in m perspective images. The goal is to recover 3D structure (line locations) and motion (camera locations) from the image measurements.

Let $\mathbf{L}_l$ be the unknown Plücker line coordinates [30] of the 3D lines, $\mathtt{P}^i$ the unknown 3×4 camera projection matrices, and $\mathbf{l}_l^i$ the measured homogeneous coordinate vectors of the image

lines, where $i = 1, \ldots, m$ labels images and $l = 1, \ldots, n$ labels lines. No point correspondences are used. An example of such line correspondences is shown in figure 1.2.

A similar image projection equation to (2.1) holds for lines and says that $\mathbf{l}_l^i$ are the projections of $\mathbf{L}_l$ up to unknown scale factors γ_l^i:

$$\gamma_l^i \mathbf{l}_l^i = \mathtt{Q}^i \mathbf{L}_l \tag{2.3}$$

where $\mathtt{Q}^i$ are the line projection 3×6 matrices of rank 3 given by

$$\mathtt{Q}^i = \begin{bmatrix} \mathtt{P}^{i2} \wedge \mathtt{P}^{i3} \\ \mathtt{P}^{i3} \wedge \mathtt{P}^{i1} \\ \mathtt{P}^{i1} \wedge \mathtt{P}^{i2} \end{bmatrix} \tag{2.4}$$

where $\mathtt{P}^{ir\top}$ are the rows of the point camera matrix $\mathtt{P}^i$, and $\mathtt{P}^{ir} \wedge \mathtt{P}^{is}$ are the Plücker line coordinates of the intersection of the planes $\mathtt{P}^{ir}$ and $\mathtt{P}^{is}$ [30, p187].

The complete set of image projections can be gathered into a matrix equation:

$$\underbrace{\begin{bmatrix} \gamma_1^1 \mathbf{l}_1^1 & \gamma_2^1 \mathbf{l}_2^1 & \cdots & \gamma_n^1 \mathbf{l}_n^1 \\ \gamma_1^2 \mathbf{l}_1^2 & \gamma_2^2 \mathbf{l}_2^2 & \cdots & \gamma_n^2 \mathbf{l}_n^2 \\ \vdots & & \ddots & \vdots \\ \gamma_1^m \mathbf{l}_1^m & \gamma_2^m \mathbf{l}_2^m & \cdots & \gamma_n^m \mathbf{l}_n^m \end{bmatrix}}_{\mathtt{S}} = \underbrace{\begin{bmatrix} \mathtt{Q}^1 \\ \vdots \\ \mathtt{Q}^m \end{bmatrix}}_{\substack{\mathtt{Q} \\ 3m \times 6}} \underbrace{[\mathbf{L}_1 \ldots \mathbf{L}_n]}_{\substack{\mathtt{L} \\ 6 \times n}} \tag{2.5}$$

where $\mathtt{L}$ and $\mathtt{Q}$ stand for structure and motion, respectively. The $3m \times n$ matrix $\mathtt{S}$ will be called the *rescaled line measurement matrix*. Only $\mathbf{l}_l^i$ are available from perspective images. Scalars γ_l^i are unknown. The task is to find scalars γ_l^i so that matrix $\mathtt{S}$ can be factorized into the matrices $\mathtt{Q} \in \mathbb{R}^{3m \times 6}$ and $\mathtt{L} \in \mathbb{R}^{6 \times n}$ such that every row of $\mathtt{Q}$ and every column of $\mathtt{L}$ as a vector, say $(v_1, v_2, v_3, v_4, v_5, v_6)^\top$, lies on the Klein quadric, i.e.

$$v_1 v_4 + v_2 v_5 + v_3 v_6 = 0 \tag{2.6}$$

which is the necessary condition for representing cameras resp. lines in Plücker coordinates.

Methods for solving projective and metric reconstructions from lines are described in sections 7.1 and 7.2, respectively.

3 The State of the Art

In this section, an overview of most relevant methods to the solved problems will be given. They are projective and metric reconstruction (sections 3.1 and 3.2, respectively), line reconstruction (section 3.3), and triangulation (section 3.4). Finally, section 3.5 presents a 2D-to-3D pipeline developed in Center for Machine Perception in Prague, which consists of several state-of-the-art components.

3.1 Projective Reconstruction

In 3D reconstruction, the situation is similar for two, three, and four uncalibrated images. 3D structure of a scene can be recovered up to an unknown projective transformation, where the camera geometry can be represented by the fundamental matrix, the trifocal, and the quadrifocal tensor, respectively. The multi-linear tensors are described, e.g., in [30].

An overview of methods for estimating the epipolar geometry is given in [31]. For non-affine situations, method [122] is invariant to Euclidean transformations of the image. Moreover, it exhibits the improved stability of previous methods for estimating the epipolar geometry, such as the preconditioning method of Hartley [31].

For any number of images, image coordinates of the projections of 3D points can be combined into a so-called *measurement matrix* (*p*7). Tomasi & Kanade [119] developed a factorization method of the measurement matrix for scene reconstruction with an orthographic camera and Sturm & Triggs [113] extended this method from affine to perspective projections by estimating

Table 3.1: Comparison of some 3D reconstruction methods from points. Lexicographical ordering was used so that (i) the importance of a criterion decreases from the first to the last column and (ii) the quality of the method decreases from top to down.

Algorithm	**camera views**		**occlusions**	**privileged data /limitation**	**depends on view ordering**
our work, sections 6.1–6.8	**N**	**perspective**	**yes**	**no**	**no**
Guilbert [27]	N	affine	yes	narrow baseline[a]	no
Fitzgibbon & Zisserman [21]	N	perspective	yes	no	yes
Avidan & Shashua [7]	N	perspective	yes	no	yes
Rother & Carlsson [97]	N	perspective	yes	reference plane	no
Urban et al. [127]	N	perspective	yes	central view	no
Heyden [35]	N	perspective	no	no	no
Tang & Hung [116, 117]	N	perspective	yes	narrow baseline	no
Mahamud & Hebert [58]	N	perspective	no	narrow baseline	no
Sturm & Triggs [113]	N	perspective	no	no	yes
Jacobs [43]	N	affine	yes	no	no
Tomasi & Kanade [119]	N	affine	yes	initial submatrix	no
Hartley & Zisserman [30]	2,3,4	affine perspective	no	no	no

[a] see end of section 6.3.4

the projective depths using multiview constraints. Another way of depth estimation is to set all depths to some initial value, for instance one, and to use some iterative process enhancing the estimate w.r.t. the reprojection error of the reconstruction. Heyden's method [35] relies on a subspace metric. Factorization is used in methods of Mahamud & Hebert [58] and Tang & Hung [116, 117]. Weaknesses of these iterative methods are (i) a (possibly) high number of iterations and (ii) no guaranty of convergence in difficult configurations. It turned out in our tests that method [58] can only be used for weak perspective or for full perspective with a good initial depth estimate. For instance, when initialized by factorization with unit depths, method [116] dit not converge in our implementation on image pair 4-5 of the St. Martin rotunda, see figure 6.3, *p*55.[1]

Occlusions present a significant problem for reconstruction. The above mentioned Tomasi & Kanade's method solves this problem under the orthographic projection but the result depends on the choice of some initial submatrix of the measurement matrix. (However, finding the largest complete submatrix in a matrix with missing elements is known to be NP-hard [43].) The method is iterative and errors may increase gradually with the number of iterations. Jacobs' method [43] improves the above approach so that no initial submatrix is needed. He combines constraints on the reconstruction derived from small submatrices of the full measurement matrix. It treats all data uniformly and is independent of image ordering.

Under the perspective projection, the occlusion problem has not yet been generally solved. Method [127] by Urban et al. is dependent on the choice of a central image, which is combined with other images in a so called "cake" configuration. Only points whose projections are contained in the central image can be reconstructed. In incremental SfM (structure from motion) algorithms, e.g., [50, 7, 91, 76], subsequent images are taken one after another and used to extend and improve the actual (projective) reconstruction. Such simple incremental approach is well suited for video-sequences, where camera moves relatively slowly and fluently. In method by Avidan & Shashua [7], consecutive fundamental matrices are "threaded" using the trifocal tensor as a connecting thread. Method by Fitzgibbon & Zisserman [21] uses a hierarchical approach: first three-view sub-sequences are estimated and iteratively registered into longer subsequences using homographies and bundle adjustment. Method by Rother & Carlsson [97] is based on having four points on a reference plane visible in all views.

So called factorization methods, e.g. [119, 113, 43], try to fill the unknown elements of the matrix of all measurements. In contrast to these, in Guilbert's method [27], affine fundamental matrices are estimated from the image measurements and affine camera matrices are estimated from the fundamental matrices.

Projective-to-metric upgrade can be done by a linear estimate of some camera intrinsic parameters (which are directly related to the image of the absolute quadric [91, equation (23)]) followed by a non-linear refinement [92, 91, 78]. Alternatively, one can use an exhaustive search for the plane at infinity by sampling the space of its possible positions [33].

Recently method of Tardif et al. [118] appeared. It is a variation on our method [65], see the end of section 6.3 for the differences.

Table 3.1 summarizes the differences among some of the named methods.

Outlier Detection

Heyden [38] presented a reconstruction method from affine images with outliers but occlusions are not handled. He extended the method into the perspective case[2]. Robust factorization

[1] At iteration 24, it was not converged yet. On the other hand, when outliers were present, it converged soon. Outliers were removed using our iterative factorization scheme. In this pair, correspondences lie almost on a flat surface, however, the EG estimation is reliable using other methods (see section 4.3).

[2] personal communication

was proposed by Aanæs et al. [5]. In an iterative scheme, a linearized version of the nonlinear problem with the perspective camera is solved. Robustness is achieved by attaching an uncertainty to individual image points, as proposed by Irani & Anandan [42]. In [51], dense correspondences from consecutive image pairs are linked for each image pixel. By keeping track of surface visibility and measurement uncertainties, it can cope with occlusions and mismatches while achieving accurate depth estimates.

3.2 Metric Reconstruction

It is known that the problem of multiview metric reconstruction can be solved in two steps: first estimate camera rotations and then translations using them.

Before the work of Nistér [77] in 2004, metric reconstruction from two views was possible only by upgrading a projective reconstruction to the metric one [31] (see section 4.2 for more references). An alternative approach was to make some assumptions about the internal camera parameters, set them to some expected values (typically square pixel, principal point in image center, focal length between 500 and 2000 for a medium image resolution[3]) and perform bundle adjustment [101]. In 2004, Nistér [77] published a closed-form solution for relative camera orientation (rotation and translation) from five points in two views. This enabled him to perform incremental metric structure-from-motion [6] in real-time.

Recently, methods minimizing the L_∞-norm of the reprojection error appeared in vision community. In our work we use Kahl's method [44] based on Second Order Cone Programming (SOCP) which is a standard technique in convex optimization. Method [44] is capable of estimating both camera translations and point positions given rotations. While [44] may fail due to a single mismatch, method of Sim & Hartley [105] cannot as it relies on translation directions between camera pairs instead of on individual point correspondences. In [104], a speed up of [44] was achieved by computing the geometry using only a few sampled data points. In [82], an efficient implementation was done using pseudoconvexity. Local optimization methods are used for more efficient computations.

Triangulation and camera resectioning using SOCP was used in incremental structure-from-motion in [49] giving good results even for forward camera motion thanks to the use of as many frames as possible for triangulation of a 3D point. (Note that optimal solutions for the camera pose estimation and the registration problems were given in [83] and [84], respectively. The approaches are based on ideas from global optimization theory, in particular, convex underestimators in combination with branch and bound technique.)

Optimal solution to the relative rotation problem in the essential matrix estimation was given in [29]. According to our best knowledge, the only method for estimating multiple camera rotations which might be possibly used for the wide baseline stereo is the work of Uyttendaele et al. [128]. In [128], differences between rotations parameterized using quaternions were nonlinearly minimized while using some additional constraints like vanishing points.

Levi & Werman [53] studied the missing data problem for uncalibrated cameras. Note that the problem is easier when camera internal calibration is known as less degrees of freedom are to cope with. Thus, more scenes can be reconstructed uniquely provided that some camera centers are not collinear (given only two-view correspondences).

Outlier Detection

Sim & Hartley [106] proved for a wide class of L_∞ problems that the set of measurements with greatest residual must contain at least one outlier. Thus, one could keep throwing out the measurements with greatest residual.

[3] from personal communication with Frederik Schaffalitzky

Degeneracies

There are several methods on distinguishing between full 3D EG and degenerate cases like planes, see, e.g., [121, 120, 48, 94, 90, 16, 23].

Constrained Camera Motion

Method by Zhong & Hung [136] is applicable on (turntable) sequences obtained by a camera undergoing circular motion. The method first computes a circular projective reconstruction of a sub-sequence and then extends the reconstruction to the complete sequence. Camera matrix and the motion parameters, i.e. the rotation angles, are computed iteratively in a way that minimizes the 2D reprojection error, similarly as in [116]. Circular projective reconstruction has a property defined in the metric space. However, it is not fully metric because the first camera matrix $\mathtt{P}^1$ is only determined up to a two-parameter family.

3.3 Line Reconstruction

It is possible to reconstruct lines in space from line correspondences in images [30]. Quan & Kanade [96] proposed a factorization based reconstruction of lines from affine images. They used a line representation that allowed them in an orthographic setup to transform the problem of finding line directions into a problem of a point factorization from 1D projective camera. Kahl and Heyden [45] proposed factorization of points, lines and conics under affine projection. They did not address the problem in the perspective setup. No such attempt has been know before 1998 [47]. Triggs' factorization method on lines [125] relies on point correspondences transfered between the corresponding lines. Recently, a promising solution to the problem was proposed by Tang et al. [115] by minimizing a geometric cost function that measures the distance of the reprojected end points of the 3D segment from the measured 2D lines on different images.

Combination of Points and Lines

Oskarsson et al. [85] compute minimal projective reconstruction from combinations of points and lines in three views. From more views, Triggs' method [125] finds point correspondences on lines which are then used in factorization of points. Lines are not used in a representation independent of points.

3.4 Triangulation

Triangulation is the problem of finding a 3D point given cameras such that the reprojection error (or its approximation) is minimized. Hartley & Sturm gave the globally optimal solution to triangulation for two views in [34], where also provided comparison with other triangulation methods, such as linear-eigen, linear-least-square, linear iterative, mid-point, and bundle adjustment methods. The optimal solution for so-called directional error in two-views was given by Oliensis [81]. The optimal solution for three views was given by Stewénius et al. [112].

First a linear method will be reviewed [34, 31]. The basic projection equation for the perspective camera is

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{\begin{bmatrix} \mathbf{p}^{1\top} \\ \mathbf{p}^{2\top} \end{bmatrix} \mathbf{X}}{\mathbf{p}^{3\top}\mathbf{X}} \tag{3.1}$$

where $\mathbf{p}^{k\top}$ are the rows of camera matrix $\mathtt{P}$ and $\mathbf{X} \in \mathbb{R}^4$ are homogenous coordinates of a 3D point. After multiplying both equations in (3.1) by the (unknown) depth, $\mathbf{p}^{3\top}\mathbf{X}$, one obtains for two images [31, $p312$]:

$$\begin{bmatrix} x\mathbf{p}^{3\top} - \mathbf{p}^{1\top} \\ y\mathbf{p}^{3\top} - \mathbf{p}^{2\top} \\ x'\mathbf{p}'^{3\top} - \mathbf{p}'^{1\top} \\ y'\mathbf{p}'^{3\top} - \mathbf{p}'^{2\top} \end{bmatrix} \mathbf{X} \quad = \quad 0_{4\times 1}. \tag{3.2}$$

This way a point observed in any number (≥ 2) of images can be triangulated.

When there is no noise in the data, the fact that the first two equations in (3.2) were reweighted by a different number than the other two takes no effect as the perfect solution always exists. However, with noisy data, the different reweighting will affect the solution. Thus, to compensate the difference in reweighting, each equation is multiplied by the reversed value of the corresponding expected depth, $\omega = \frac{1}{\lambda_{\text{exp}}}$, $\omega' = \frac{1}{\lambda'_{\text{exp}}}$:

$$\begin{bmatrix} (x\mathbf{p}^{3\top} - \mathbf{p}^{1\top})\,\omega \\ (y\mathbf{p}^{3\top} - \mathbf{p}^{2\top})\,\omega \\ (x'\mathbf{p}'^{3\top} - \mathbf{p}'^{1\top})\,\omega' \\ (y'\mathbf{p}'^{3\top} - \mathbf{p}'^{2\top})\,\omega' \end{bmatrix} \mathbf{X} \quad = \quad 0_{4\times 1}. \tag{3.3}$$

The expected depths, λ_{exp} and λ'_{exp}, can be initialized to ones or to some better estimate if available. Then, system (3.3) can be iteratively solved for $\mathbf{X}$ while resetting weights ω and ω' using the updated $\lambda_{\text{exp}} = \mathbf{p}^{3\top}\mathbf{X}$, $\lambda'_{\text{exp}} = \mathbf{p}'^{3\top}\mathbf{X}$ till the solution changes. This process converges in a few iterations.

Unfortunately, it turned out in our experiments that this iterative process does not attain the minimum which would be attained by a non-linear descent (bundle adjustment) in the $\mathbf{X}$ parameters. Since some iteration, errors in the expected depth estimates prevent from obtaining a better solution.

It has been shown in [112, section 6.4] for three views that the linear solution followed by a non-linear descent is not trapped into a local minimum. Even though the iterative process described above is worse than the non-linear descent, it may be worth using for its computational simplicity.

Note that solutions of systems (3.2) and (3.3) do not depend on normalization of the image data by a 3×3 homography restricted to translation and the same scale change in both x- and y-directions.

3.5 3D Model from Unorganized Images

In this section[4], we will show how the methods for multiview reconstruction described in chapters 5 and 6 can be used. They are used as a joint connecting the wide-baseline sparse matching on image pairs [69] and the dense [52, 12] stereo matching algorithms developed in Center for Machine Perception at Czech Technical University (CTU) in Prague into an automated multiview reconstruction pipeline. The pipeline is summarized in algorithm 1, *p*29.

Individual processing steps are arranged into an automatic pipeline. The input is an unorganized set of images which overlap in some image pairs, and the output is 3D information about object surfaces. Importantly, camera calibration and parameter estimation is not needed during the acquisition.

The 3D reconstruction example uses the sculpture *Ecoute* (meaning 'listen' in English) by Henri de Miller from 1986 which can be seen outside St. Eustache's church in Paris. Altogether, 26 images of the sculpture were captured and used for reconstruction. The resolution of each image was 1136×852 pixels. The data set is shown in figure 3.1.

An attempt is made to establish sparse correspondences across all image pairs. Pairwise image matching is performed with Local Affine Frames [70] constructed on MSER regions, LaplaceAffine and HessianAffine [75] interest points. The methods used to match images [70, 75] can handle large changes in scale and brightness. Examples of regions used are shown in figure 3.2. An epipolar geometry unaffected by a dominant plane is found using [16] (section 4.3), see figure 3.2.

The example shown in this section was reconstructed using method [66] (section 6.7). Nevertheless, any method described in sections 6.1–6.8 can be used. The estimated camera positions and orientations are depicted in figure 3.3.

The next step is pairwise image rectification [72, 71], which improves matching efficiency. We use Hartley's method [32]. After this step, the epipolar lines correspond to image rows. Pairs for dense matching are selected based on the mutual locations of the cameras, as described in [17]. An example of one such pair is shown in figure 3.4. Dense correspondences can be sought on a line-by-line basis between the left and the right images.

Having rectified the image pairs, a disparity map can be computed using a dense stereo algorithm. Stratified Dense Matching [52, 12] is used. The algorithm has a very low mismatch rate, it is fast, robust, and accurate, and does not need any parameters which are difficult to set. The output from the matching algorithm is a disparity map for each image pair admitted for dense matching. By least squares estimation using an affine distortion model, the disparity

[4]Parts of this section appeared in [17, section 2], [46, section 2.1] and [109, section 12.5].

Figure 3.1: Ten of twenty six input images of Henri de Miller's sculpture Ecoute. All twenty six images are used for reconstruction. *Courtesy Ondřej Chum, CTU.*

Figure 3.2: Detected maximally stable extremal regions, LaplaceAffine, and HessinanAffine interest points. These structures constitute salient features for finding wide baseline stereo epipolar geometry (top row) Matched subset of the regions which constitute inliers to the epipolar geometry found by RANSAC (bottom row)

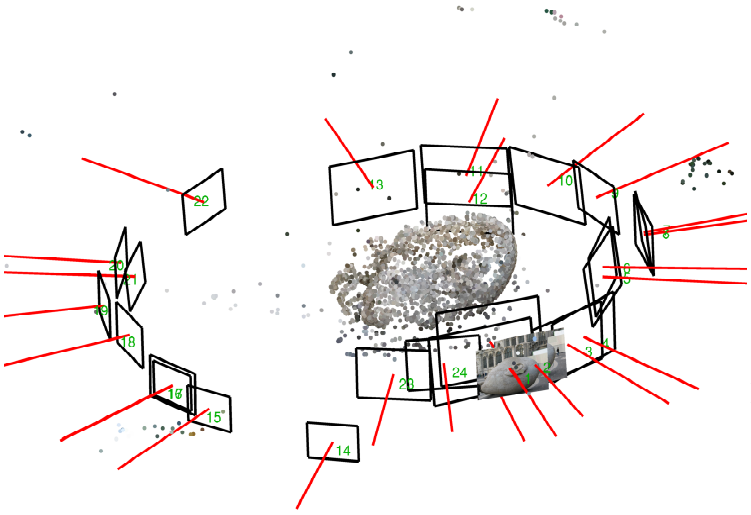


Figure 3.3: Result of automated camera calibration for all views used in reconstructing Miller's Ecoute sculpture.

Figure 3.4: An example of a rectified image pair and a disparity map computed using dense stereo algorithm [52]. The disparity, encoded in color, corresponds to scene depth.

Figure 3.5: A cloud of 3D points aggregates measurements provided by all 26 input images. Only 10% of all 3D points is shown (left) 3D points represented by fish-scales without texture (middle) and with appropriate texture (right)

maps achieve sub-pixel resolution [99]. An example of a resulting disparity map is shown in figure 3.4. (Disparity is inversely proportional to scene depth.)

The disparity maps are used to reconstruct the corresponding 3D points. The union of the points from all disparity maps forms a dense point cloud as can be seen in figure 3.5left.

An efficient way of representing distributions of points is to use fish-scales [100]. Fish-scales are local covariance ellipsoids that are fitted to points by the k-means algorithm. They can be visualized as small round discs. Each fish-scale encodes a 3D point and an estimate of the plane tangent to the cloud at this point. A collection of fish-scales approximates the spatial density function of the measurement in 3D space. Figure 3.5middle shows the reconstructed set of fish-scales without texturing. The view would reveal even small errors in the reconstruction of local surface orientation. Figure 3.5right then shows a textured model where the texture for each fish-scale is taken from the image which has the best resolution over the fish-scale. Texture from individual images is not corrected for camera exposure time.

In [95], a system for 3D reconstruction from video was proposed. Our system differs in two aspects: (i) the 3D reconstruction can be obtained from wide baseline images (chapter 6) and (ii) the scene can be automatically segmented into objects [46], see figure 3.6. In [95], images were rectified using method [93]. Multi viewpoint stereo [51] was used, which can be seen as an alternative to fish-scales. In [51], dense correspondences from consecutive image pairs are linked for each image pixel. By keeping track of surface visibility and measurement uncertainties, it can cope with occlusions and mismatches while achieving accurate depth estimates. Recently, graphics cards were used for real-time implementations of algorithms for, e.g., dense stereo [133,

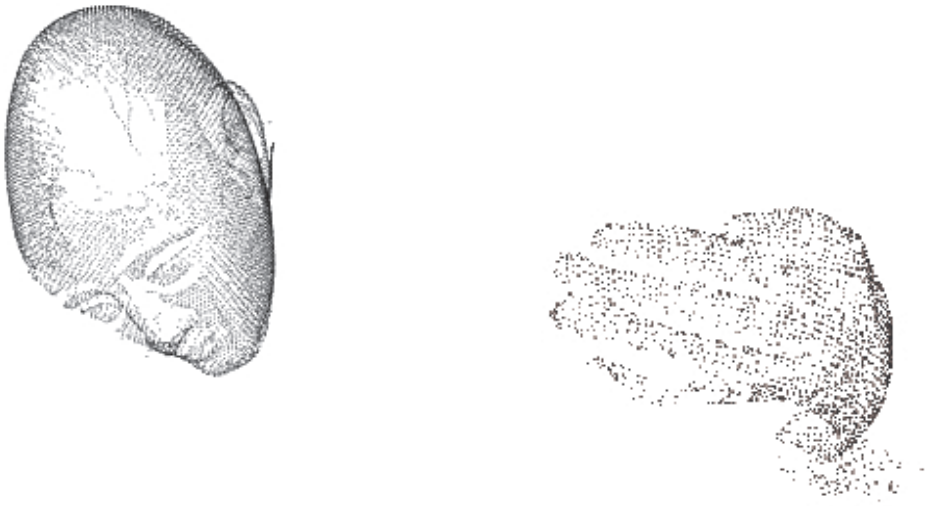


Figure 3.6: Orientation-topology output: the two largest 2D components of the cloud from figure 3.5middle. Together they contain 12182 points (fishscales) out of the total 12692 points.

24] and multiview stereo [134].

4 Geometry of Multiple Views

In this section, several problems in geometry of several views will be dealt with. Our interpretation of depths in the epipolar geometry is given in section 4.1. Section 4.2 introduces a method for relative pose of two cameras without knowledge of the focal length ratio. Our robust estimation of the epipolar geometry using RANSAC is explained in section 4.3. The method can deal with m-n tentative correspondences and dominant planes. Speed is enhanced using local optimization [13]. Image triplets provide stronger constraints on 3D reconstruction and thus higher accuracy can be achieved compared to using image pairs only. On the other hand, the data used is larger and it needs a special treatment. It will be explained together with mismatch identification and robust estimation of a camera triplet in section 4.4. Our linear method for triangulation in any number of views is presented in section 4.5. Robust L_∞-norm estimation is provided in section 4.6.

4.1 Depths in Epipolar Geometry

The relation between (projective) depths in different views was described by Sturm & Triggs in [113, equation (2)]. Figure 4.1 explains this relation geometrically. Our derivation is given below.

Let β and β' denote coordinate systems of the first and other camera (image measurements), respectively. Consider the triangle relating the two camera centers and a 3D point. The triangle is obtained after proper rescaling the two image measurements (vectors $\mathbf{x}_\beta$ and $\mathbf{x}'_\beta$ expressed in the same basis, β) by depths λ and λ' and the epipole by τ:

$$\lambda \mathbf{x}_\beta \quad = \quad \tau \mathbf{e}_\beta + \lambda' \mathbf{x}'_\beta. \qquad / \ \mathbf{e}_\beta \wedge \tag{4.1}$$

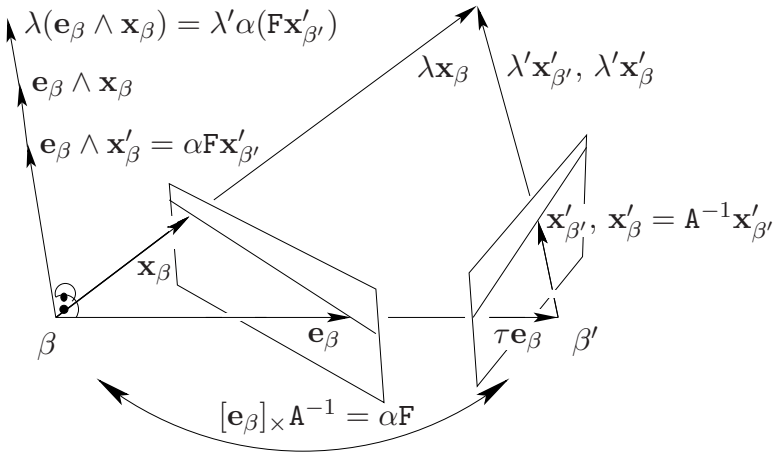


Figure 4.1: Depths in epipolar geometry – geometrical explanation.

After multiplying both sides of equation (4.1) by the cross-product with $\mathbf{e}_\beta$, one obtains:

$$\lambda(\mathbf{e}_\beta \wedge \mathbf{x}_\beta) \quad = \quad \lambda'(\mathbf{e}_\beta \wedge \mathbf{x}'_\beta). \tag{4.2}$$

Notice that $\mathbf{x}'_\beta$ in the above equations is expressed in the coordinate system of the first camera but it was measured in the other system. Let $\mathtt{A}$ maps vectors from basis β to β'. Then, $\mathbf{x}'_\beta = \mathtt{A}^{-1}\mathbf{x}'_{\beta'}$ and

$$\mathbf{e}_\beta \wedge \mathbf{x}'_\beta \quad = \quad \mathbf{e}_\beta \wedge \mathtt{A}^{-1}\mathbf{x}'_{\beta'} \quad = \quad [\mathbf{e}_\beta]_\times \mathtt{A}^{-1}\mathbf{x}'_{\beta'} \quad = \quad \alpha(\mathtt{F}\mathbf{x}'_{\beta'}). \tag{4.3}$$

The last equality follows from relation $[\mathbf{e}_\beta]_\times \mathtt{A}^{-1} = \alpha\mathtt{F}$ [31, equation (9.1), $p244$]. Here, the arbitrary scale of $\mathtt{F}$ is expressed using scale α. Using equations (4.2) and (4.3) one obtains [113, equation (2)]:

$$\lambda(\mathbf{e}_\beta \wedge \mathbf{x}_\beta) \quad = \quad \lambda'\alpha(\mathtt{F}\mathbf{x}'_{\beta'}).$$

Geometrical meaning of all used terms can be seen in figure 4.1.

4.2 Relative Pose without Focal Length Ratio

A minimal solution for relative pose of two cameras has been given for known focal lengths (five-point algorithm by Nistér [77]), for unknown focal constant length (six-point algorithm by Stewénius et al. [111]). This section introduces a minimal solution when no knowledge of the two focal lengths is available.

The difference from [111] is that also the ratio between the two focal lengths is unknown. This one more unknown enforces adding one more point in the minimal solution, thus it is a seven point algorithm. Note that a seven point algorithm for uncalibrated cameras is known for a long time [31]. It provides one or three solutions depending on the number of real roots of a cubic polynomial. In this work we make use of the uncalibrated seven-point algorithm instead of making a complicated analogy of [111] for the one more unknown. The new technique simply (i) obtains one or three solutions from the uncalibrated seven-point algorithm and (ii) enforces the additional constraint on the camera intrinsics, namely the transformation from the fundamental to the essential matrix (the equation before (6) in [111]).

A review of relative pose algorithms is given in [110]. Our algorithm solves the same problem as [37]. It seems that both solutions have the same complexity. Compared to [37], our solution seems to be easier to understand.

The algorithm can be used in RANSAC to filter out samples contaminated by a mismatch. Although (uncalibrated) epipolar geometries may be obtained from contaminated samples, using the additional constraint on the essential matrix, many contaminated samples are withdrawn from further processing. As this is done early, a speedup in RANSAC is achieved.

One possible application of the method is on the images downloaded from the internet (Google images [108]) which either have no information about focal length in their JPEG EXIF header or, if available, it is difficult to relate the focal lengths given by various cameras and optics.[1]

The New Method

Once the camera intrinsics are known up to the two focal lengths, assume that image coordinates are transformed so that the two camera internal calibrations are:

$$\mathtt{K} \quad = \quad \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \mathtt{K}' \quad = \quad \begin{bmatrix} f' & 0 & 0 \\ 0 & f' & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{4.4}$$

[1] This experiment has not been done so far but we expect at least as good results with method [67] (section 6.8) as those in [108].

where f and f' are the unknown focal lengths.

From the minimum configuration of seven points, one or three fundamental matrices (uncalibrated EG) are obtained [31]. Each fundamental matrix, $\mathtt{F}$, is tried to be transformed to the essential matrix, $\mathtt{E}$, by (the equation before (6) in [111]):

$$\mathtt{E} \quad = \quad \mathtt{K}'^{\top}\mathtt{F}\mathtt{K}. \tag{4.5}$$

As in [111], the following cubic constraints on the essential matrix will be used:

Theorem 1 *[88] A real non-zero 3×3 matrix $\mathtt{E}$ is an essential matrix iff it satisfies the equation*

$$2\mathtt{E}\mathtt{E}^{\top}\mathtt{E} - tr(\mathtt{E}\mathtt{E}^{\top})\mathtt{E} \quad = \quad 0. \tag{4.6}$$

Set (analogically to equation (6) in [111])

$$\mathtt{P} \quad = \quad f^{-1}\mathtt{K} \quad = \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & f^{-1} \end{bmatrix}, \qquad \mathtt{P}' \quad = \quad f'^{-1}\mathtt{K}' \quad = \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & f'^{-1} \end{bmatrix},$$

then equation (4.6) is equivalent to

$$\begin{aligned} 2\mathtt{P}'\mathtt{F}\mathtt{P}\mathtt{P}^{\top}\mathtt{F}^{\top}\mathtt{P}'\mathtt{P}'\mathtt{F}\mathtt{P} - tr(\mathtt{P}'\mathtt{F}\mathtt{P}\mathtt{P}^{\top}\mathtt{F}^{\top}\mathtt{P}')\mathtt{P}'\mathtt{F}\mathtt{P} \quad &= \quad 0 && (4.7)\\ \Leftrightarrow \quad 2\mathtt{F}\mathtt{P}\mathtt{P}\mathtt{F}^{\top}\mathtt{P}'\mathtt{P}'\mathtt{F} - tr(\mathtt{P}'\mathtt{F}\mathtt{P}\mathtt{P}\mathtt{F}^{\top}\mathtt{P}')\mathtt{F} \quad &= \quad 0 && (4.8)\\ \Leftrightarrow \quad 2\mathtt{F}\mathtt{P}^2\mathtt{F}^{\top}\mathtt{P}'^2\mathtt{F} - tr(\mathtt{F}\mathtt{P}^2\mathtt{F}^{\top}\mathtt{P}'^2)\mathtt{F} \quad &= \quad 0. && (4.9) \end{aligned}$$

Notice f^{-1} and f'^{-1} only appear in even powers in the above set of polynomial equations and hence one can set $p = f^{-2}$ and $q = f'^{-2}$.

The nine equations (4.9) can be written as $\mathtt{A}\mathbf{X} = 0$ where $\mathtt{A}$ is a 9×4 matrix of scalars and $\mathbf{X}$ is a vector of monomials

$$\mathbf{X} \quad = \quad [pq, p, q, 1]^{\top}.$$

To obtain a non-trivial solution, which is needed as the last component of the $\mathbf{X}$ vector equals one, rank of the $\mathtt{A}$ matrix must be at most three. In all our experiments, the $\mathtt{A}$ matrix had rank three. It is perhaps thanks to the singularity of the fundamental matrix $\mathtt{F}$. The solution can be obtained as the null-space of the $\mathtt{A}$ matrix, e.g., using MATLAB's NULL, $\mathbf{x} = \text{NULL}(\mathtt{A})$, and proper normalization:

$$p \quad = \quad \frac{\mathbf{x}_2}{\mathbf{x}_4}, \qquad q \quad = \quad \frac{\mathbf{x}_3}{\mathbf{x}_4}.$$

One can use a more efficient computation using Gauss-Jordan elimination. One has to take care that pivoting is done on nonzero elements (from numerical reasons, with absolute value larger than, e.g., 10^{-14}). [2]

The focal lengths are computed as $f = \frac{1}{\sqrt{p}}$ and $f' = \frac{1}{\sqrt{q}}$. To obtain calibrated cameras, both focal lengths have to be real and thus p and q have to be positive. Relative pose ($\mathtt{R}$, $\mathbf{t}$) can be obtained by decomposing the essential matrix [31] obtained using equation (4.5).

We have not studied behaviour of this method on critical point configurations [89]. In the $\mathtt{A}$ matrix, only multiplications of entries of the fundamental matrix appear. Thus, there are no divisions like in [9]. Moreover, as only elements of the fundamental matrix are used in the computation, eventual problems with representation of the second projection matrix cannot happen like in [37].

[2] Any three linearly independent rows give the only one solution. It turned out that (perhaps always, not proven) rows 1, 2, and 4 are linearly independent (rows 1, 2, 3 are linearly dependent).

4.3 RANSAC on Epipolar Geometry

An overview of methods for estimating the epipolar geometry (EG) is given in [31]. When using point neighborhood, only three [14] or two [86] correspondences between regions are needed for uncalibrated or calibrated cameras, respectively. A provably optimal model verification in RANSAC was given in [68].

Several important techniques not mentioned in [31] need to be incorporated in RANSAC to find a correct EG of a difficult image pair.

1. *Local optimization.* Local optimization (LO) [13] is applied the same way as in DEGENSAC [16] except that instead of uncalibrated cameras (eight-, seven-point algorithms), calibrated cameras are used (seven-point (section 4.2), six-point, and five-point algorithms). Note that all k-point algorithms can use any number, p, of points such that $p \geq k$ by using the rightmost singular vectors (corresponding to the smallest singular values) instead of the nullspace of the matrix formed from point correspondences [31, equation (11.3), p279].

2. *A dominant plane.* In [66] (section 6.7), first an EG for an uncalibrated camera unaffected by a dominant plane was found using DEGENSAC [16]. Then, the found inliers were used as the pool for drawing samples in calibrated RANSACs [111, 77].

   In our current implementation, the seven-point (section 4.2), six-point and five-point algorithms [111, 77] are used instead of the seven-point algorithm in [16]. The H-degeneracy test is performed on two triplets only (as described for image triplets in section 4.4).

   Local optimization (LO) [13] is used in robust plane estimation. When it is detected that the sample is H-degenerate [16], i.e. at least five points in the sample are related by a homography, first plane plus parallax algorithm [41, 31] for uncalibrated calibrated cameras is applied the same way as in DEGENSAC. Then, the calibrated RANSACs are used on the inliers found for the uncalibrated case. When a plane was detected, the samples in calibrated RANSACs are drawn so that some (three) points lie in the plane and the rest out of the plane.[3]

   It turned out that sometimes the calibrated RANSAC finds much larger support compared to the LO with uncalibrated cameras, see figure 4.2. This may seem surprising as one could expect the calibrated LO to find smaller support because the geometry must satisfy more constraints. On the other hand, the camera calibration serves as a guided matching thanks to which the remaining matches can be localized more precisely.

3. *m-n tentative correspondences.* Based on correlation between regions in the first and the other image, so-called *tentative correspondences* are established [69] before running RANSAC. In the classical RANSAC, only one-to-one (1-1) tentative correspondences are considered meaning that for each point in the first image there is exactly one corresponding point in the other image. However, when there are many similar regions, it may be impossible to determine the most likely correspondence among the multiple similar ones. In such a case, it is better to consider all such m-n tentative correspondences and to postpone the decision about the correct correspondence to the geometry test in RANSAC.

   On one hand, the number of m-n tentative correspondences is larger, which causes RANSAC to take more time, on the other hand, they contain more correct correspondences. Note that the few extra inliers may be crucial for finding an EG on difficult image pairs.

   In our implementation of the m-n RANSAC, only the following necessary conditions on the drawn samples and the accepted residuals are ensured. Note that there can be done much

[3] This ensures that never all chosen points lie in the plane, which is important especially for the six-point algorithm [111].

Figure 4.2: Local optimization (LO) [13] using the uncalibrated cameras found only 38 inliers (left) whereas LO exploiting the calibrated six-point [111] RANSAC found 181 inliers (right). Image pair 70-71 of the Frauen Kirche scene is shown, four iterations of the LO were performed. Matching of this pair is quite challenging due to many repetitive windows on the buildings. Some mismatches were accepted.

more in reasoning about which correspondences should be sampled more often. One such attempt has been done in [135].

a) **Sampling**: The drawn sample may contain only 1-1 correspondences.
This can be easily achieved by choosing some correspondence in random and removing all correspondences incident with the correspondence. This is repeated until all correspondences needed to describe the model are drawn.

b) **Verification**: There may be no m-n correspondence among the inliers.
Residuals of all (both m-n and 1-1) correspondences are computed. Then, if there are more points in the other image corresponding to some point in the first image, the one with the lowest residual is accepted and the remaining ones are removed. If there are more matches with equal minimal residuals, all are accepted as unclear. At the end, if some correspondences are unclear, the above procedure is repeated again but with swapped images to solve for the remaining ambiguities. Note that the residuals above the RANSAC threshold can be removed before the verification step.

Examples of scenes with many repetitive structures whose reconstruction seem to be impossible using 1-1 RANSAC are the Raglan and the Zwinger scenes, see section 6.8.

As the estimation of the geometry model (the essential matrix) using the five- and six-point algorithms is computationally demanding, it is advantageous to remove some samples using a simpler test, such as a weaker orientation constraint [130].

4.4 Using Image Triplets

There are several important aspects of using image triplets one needs to cope with.

1. *Data size.* Number of image triplets can be huge if many images have large overlaps. Thus, a strategy for choosing some representative ones may be needed due to memory and and time issues. Two different strategies were proposed: (i) using a minimal set of image triplets in [65] (section 6.3) and (ii) strengthening of the most important EGs in [66, section 4.1] (section 6.7.5).

For representing the constraint given by an image triplet, it is sufficient to use four points only, see section 6.8.3.

2. *Outlier removal.* Several ways of outlier removal have been used: (i) iterative factorization with rejecting points with large reprojection errors [65, section 2.3] (section 6.3.3) (ii) a rough reconstruction of the image triplet from pair-wise reconstructions using algorithm [66, section 3.2] (section 6.7.2) followed by four point RANSAC with model obtained using BA initialized by the rough estimate (iii) six-point RANSAC using uncalibrated cameras [102] (see lower).

   An advantage of approach (ii) over approach (iii) is that metric reconstruction is obtained thanks to exploiting known camera internals. However, it is more time consuming. One can use an estimate of all rotations from all pairwise reconstructions and estimate the three-view reconstruction of the four points using [44]. However, this estimate is corrupted by the error in the focal length (section 5.3). Thus, metric BA with variable focal length is needed. It can be used in conjunction with local optimization (LO) [13] or a robust kernel on reprojection error, see section 5.4.2.

   On the contrary, the uncalibrated six-point RANSAC [102] (iii) gives model which is precise on the sampled six points. However, projective BA (section 5.4.1) is needed (with LO or a robust kernel).

3. *A dominant plane.* To correctly handle the presence of a dominant plane, one should use a variation on DEGENSAC [16] for three views. For each sample, it is necessary to detect when the six (or a subset of five) points lie on a plane.

   Note that local optimization [13] used without the degeneracy test may not help when there are only a few correct off-plane points compared to mismatches. Even if they are, by chance, incorporated into early iterations with loose thresholds, the amount of mismatches can cause that the model fits to them rather to the few off-plane correct points. In further iterations, the correct correspondences may stop being inliers w.r.t. the (wrong) model.

   Detection of H-degenerate Samples in Three Views (variation on [13, section 3]). From the six point correspondences, the model is estimated as three projection matrices of uncalibrated cameras using [102]. Any of $\binom{6}{5} = 6$ five-tuples contains at least one of triplets $\{1, 2, 3\}$ and $\{4, 5, 6\}$. Hence, at most two homographies have to be tested.

   A bit weaker constraint can be employed. Instead of a single three-view model, three two-view models (F or H) corresponding to the three image pairs are searched for. The worst thing that may happen is that the final trifocal tensor will not fit all the data together but only pair-wise. Nevertheless, it is no problem to reject such points at the very end (as it is done in DEGENSAC [16, step 6 in algorithm 1]).

   One could claim that only one image pair is needed to detect a plane. However, this would work only in case when all cameras have distinct centers. If two camera centers coincide (they belong to a panorama), all their points will fit a plane. However, the plane does not exist in 3D, it is just a mapping of images. The true 3D plane can be distinguished using the third image, if it has a distinct center. If all three camera centers coincide, the 3D plane is indistinguishable. All points would be (perhaps incorrectly) classified as in-plane however correctly classified as inliers. [4]

[4] When all points lie in a plane, some representative of an infinite class of solutions for the trifocal tensor will be found. In the case of cameras calibrated up to an overall focal length, this seems to correspond to that any focal length is possible, as in the two-view case [111].

On the contrary, only two (instead of three) image pairs are sufficient to detect the plane. However, in our implementation, all three image pairs are used due to symmetry and to increase reliability in decisions such as degenerate/non-degenerate sample when small numbers close to machine accuracy are compared.

Compared with [55], our approach uses three homographies corresponding to the three image pairs instead of one virtual homography. The three fundamental matrices and homographies are estimated from a few noisy measurements and thus they are not consistent. Nevertheless, enforcing mutual consistency is not needed. Instead, a three-view correspondence is considered to lie on the plane if all the three pair-view correspondences are related by the corresponding homographies (within some accuracy). Otherwise, it is considered to lie off the plane.

Note that the H-degeneracy test has not been applied so that camera internal calibration would be exploited.

The H-degeneracy detection in three views has not been published yet as most problems were solved using two views only in [67] (section 6.8).

4.5 Triangulation with Depths

This section brings a new linear solution to the triangulation problem using projective depths, which is applicable for any number of views. See section 3.4 for an overview of known triangulation methods.

The constraint that reprojections should be close to the measured data can be expressed using (projective) depths:

$$\left.\begin{array}{rcl} \lambda\mathbf{x} & = & \mathtt{P}\mathbf{X} \\ \lambda'\mathbf{x}' & = & \mathtt{P}'\mathbf{X}. \end{array}\right\} \tag{4.10}$$

In fact, equation (4.10) is a variation on the basic projection equation for the perspective camera (3.1), *p*12 with λ supposed to be close to $\mathbf{p}^{3\top}\mathbf{X}$.

The image coordinates should be normalized to be close to one [113]. Then, the error minimized in equations (4.10) solved in the least squares sense can better approximate the reprojection error compared to the purely algebraic error in equations (3.3), *p*13. If all the depths are close to equal, then equations (4.10) minimize a good approximation to the reprojection error [31, *p*446]. This proves to be fruitful when more than two views are used. For instance, in three-view triangulation, equations (4.10) often provide a lower reprojection error than the Sampson's approximation [31], which is far better than minimizing the algebraic error. In two-view triangulation, the Sampson's approximation [31] was always better in our experiments than the triangulation using depths.

Similarly to equations (3.3), equations in (4.10) can be reweighted by the inverse of the expected depth to compensate the difference in reweighting the equations by λ and λ'.

In our pipeline, for three views triangulation is performed using both the Sampson's approximation [31] and equations (4.10) and the result with the smaller error is used.

4.6 Robust L_∞-norm Estimation

Recently, methods for minimizing the L_∞-norm, i.e. the maximum, error appeared in computer vision [44, 49, 105, 106]. This section brings an attempt to make Kahl's method [44] more robust w.r.t. mismatches. An alternative approach was given by Sim & Hartley [106] in which outliers are iteratively removed as the data points with the largest residua. In contrary to this,

our technique tries to cope with all data at the same time as the Kahl's method [44] does. We formulate a relaxation of the Kahl's method that tries to prevent mismatches from harming the solution.

Robustness is achieved by adding an "error" term to each cone constraint corresponding to the reprojection error of an image point. Our task minimizes the sum of the error terms. Thus, only one minimization problem is solved instead of many feasibility problems solved in the bisection method [44].

The original SOCP feasibility problem (4) in [44]

$$\begin{array}{rccll} \text{find} & x & & & \\ \text{subject to} & |||[f_{1p}(x), f_{2p}(x)]||| & \leq & \gamma\lambda_p(x) & \\ & \lambda_p(x) > 0 & & & \forall p \end{array} \tag{4.11}$$

(index p goes via all measured image points) is modified to

$$\begin{array}{rccll} \min & \sum_{\forall p} \alpha_p & & & \\ \text{subject to} & |||[f_{1p}(x), f_{2p}(x)]||| & \leq & \gamma\lambda_p(x) + \alpha_p\lambda_{p,\text{exp}} & \\ & \lambda_p(x) > 0 & & & \\ & \alpha_p \geq 0 & & & \forall p \end{array} \tag{4.12}$$

where $f_{1p}(x)$, $f_{2p}(x)$, and $\lambda_p(x)$ are affine functions in x, γ is an upper bound on the reprojection error (see more details in [44]), $\lambda_{p,\text{exp}}$ are the expected depths, and α_p are auxiliary variables. If the p^{th} measured image point is a mismatch, the inequality condition in (4.12) may still be satisfied even for a low threshold γ thanks to that the α_p variable can become greater than zero. To ensure that this relaxation is not used on correct image points, the sum of all auxiliary variables $\sum_{\forall p} \alpha_p$ is minimized in the task.

All data are used with expected depths $\lambda_{p,\text{exp}}$ set to ones. If all α_p are zeros, there is no mismatch and the process ends with a solution equivalent to the original task (4.11). When desired, problem (4.12) may be solved again with $\lambda_{p,\text{exp}}$ set to the estimate of $\lambda_p(x)$, which can be possibly iterated. In this process, the expected depths, $\lambda_{p,\text{exp}}$, keep improving towards the estimated depths, $\lambda_p(x)$. The process converged in our experiments. The stopping criterion can be, e.g., when the difference between the expected and the estimated depths lowers below some threshold.

Although mismatches influence the solution of problem (4.12), they do not harm it completely thanks to that the second order cone is relaxed to absorb also mismatches. Formulation (4.12) can be applied for both triangulation and multiview reconstruction with known rotations [44].

Handling Mismatches

Points with large reprojection errors are likely to be mismatches. Their influence can be further easily diminished by reweighting the criterion function:

$$\min \quad \sum_{\forall p} \omega_p \alpha_p \tag{4.13}$$

where weight ω_p representing uncertainty of the estimated point can be set using, e.g., the Gaussian kernel to

$$\omega_p = e^{-\frac{\alpha_p^2}{\gamma^2}}.$$

Implementation

In implementation of [44], constraints $\lambda_p(x) > 0$ do not have to be given explicitly as they follow from condition $|||[f_{1p}(x), f_{2p}(x)]||| \leq \gamma\lambda_p(x)$. However, in (4.12), the positivity of $\lambda_p(x)$ does not

follow from $|||[f_{1p}(x), f_{2p}(x)]|| \leq \gamma\lambda_p(x) + \alpha_p\lambda_{p,\text{exp}}$ due to the additional term with α_p. Thus, constraints $\lambda_p(x) > 0$ have to be named explicitly. To prevent numerical difficulties, a threshold on a minimum depth is applied, $\lambda_p(x) > \lambda_{min}$ (0.1). The coordinate system was fixed the same way as in [44].

This method has not been published before and is not used currently as a very reliable mismatch removal was developed [67] (section 6.8).

5 Pipeline from Unorganized Images to a 3D Model

This chapter describes several techniques used in the CMP reconstruction pipeline from unorganized images to a sparse 3D model, which is summarized in algorithms 1 and 2.

1. **Image matching** (section 5.1). In image pairs (not necessarily all), regions of interest are matched [70] and verified using two-view geometry constraints, see figure 3.2, *p*15. Robust RANSAC estimation on m-n matches is used (section 4.3).
2. **Joining Matches into Tracks** (section 5.2). The measurement matrix (MM) (*p*7) is build by merging pairwise matches from step 1.
3. **Focal Length Estimation** (section 5.3). The overall scale of focal length is estimated using all measurements in the MM. Using the new focal length, all pairwise geometries are reestimated.
4. **Multiview Reconstruction Estimation**. Using the pairwise geometries and the MM, sparse multiview reconstruction is estimated by any method from chapter 6. The best current method is summarized in algorithm 2.
5. **Dense Model Estimation**. Image pairs suitable for dense stereo are chosen, see section 5.5. In each chosen image pair, the two images are epipolarly rectified [32, 71] and densely matched [12, 52] (figure 3.4, *p*16). Fish-scales [100] are fit to the resulting point cloud (figure 3.5, *p*16). See section 3.5 for more details.

Algorithm 1: Pipeline from unorganized images to a 3D model.

Building a multiview reconstruction can be started either from a given measurement matrix, from pairwise reconstructions, or from plain images. When pairwise matches are available, the MM can be obtained by *merging the pairwise matches into tracks*, which will be explained in section 5.2. When only images are given, it would be best to match all $\binom{m}{2}$ image pairs. However, as the number of pairs grows quadratically with the number of images, it would take too much time to process hundreds or thousands of images. Therefore, section 5.1 introduces a new heuristics thanks to which not all image pairs need to be matched to obtain a reconstruction of comparable quality.

The remaining sections provide a detailed insight into the focal length calibration from images (section 5.3), several variants of the bundle adjustment (section 5.4) and detection of image pairs with very short baseline (section 5.5).

5.1 Image Matching

Images captured with the intention of making a 3D reconstruction are often not taken entirely arbitrarily. Usually, while capturing images, a person moves with a camera along a more or less continuous path. Thus, the image positions have roughly a sequential form, meaning here that

Input: Pairwise, ij, metric reconstructions represented by a camera pair, $\mathtt{P}^{ij}$, $\mathtt{P}'^{ij}$, 3D points, $\mathtt{X}^{ij}$, and measurement matrix, $\mathbf{x}$ ($p7$).
Output: Cameras, $\mathtt{P}^i$, $i = 1, \ldots, m$, and inlying points, $\mathtt{X}_p$, $p \in \{1, \ldots, n\}$, projecting close to $\mathbf{x}$.

1. **Mismatch Suppression and Data Compression** (section 6.8.3). In each ij-reconstruction, fit Gaussian to the columns of the rescaled measurement matrix, $\begin{bmatrix} \mathtt{P}^{ij} \\ \mathtt{P}'^{ij} \end{bmatrix} \mathtt{X}^{ij}$, normalized as described in section 6.8.3. Remove the most likely mismatches as $\epsilon = 25\%$ of the most distant points from the center of the Gaussian in the Mahalanobis distance given by the auto-covariance matrix of the Gaussian (figure 6.23). From the remaining points, choose the most distinct four points (figure 6.25) for representing the ij-reconstruction using algorithm 6, $p103$.

2. **Relative Rotations**. For each ij-reconstruction, estimate its relative rotation as $\mathtt{R}^{ij} = \mathtt{R}'\mathtt{R}^\top$, where $\mathtt{P}^{ij} = \mathtt{K}\mathtt{R}[\mathtt{I}| - \mathbf{t}]$ and $\mathtt{P}'^{ij} = \mathtt{K}'\mathtt{R}'[\mathtt{I}| - \mathbf{t}']$ are the camera decompositions [31, equation (6.11)].

3. **Rotation Registration**. Estimate approximate rotations, $\hat{\mathtt{R}}^i$, as $\hat{\mathtt{R}}^i = [\mathbf{r}_1^i \mathbf{r}_2^i \mathbf{r}_3^i]$ where $\mathbf{r}_1^i$, $\mathbf{r}_2^i$, $\mathbf{r}_3^i$ are the best three linearly independent least squares solutions to system

$$\mathbf{r}^j - \mathtt{R}^{ij}\mathbf{r}^i = 0_{3\times 1} \quad \text{for all } ij. \tag{5.1}$$

   Obtain rotations, $\mathtt{R}^i$, as the rotations which are closest to the approximate rotations in the Frobenius norm as $\mathtt{R}^i = \mathtt{U}\mathtt{V}^\top$ where $\hat{\mathtt{R}}^i = \mathtt{U}\mathtt{S}\mathtt{V}^\top$ is the SVD factorization.

4. **Translation registration**. Estimate global camera translations, $\mathbf{t}^i$, using method [44] applied on point quadruplets representing each ij-reconstruction (see step 1) and known global camera rotations, $\mathtt{R}^i$. Compose cameras as $\mathtt{P}^i = \mathtt{K}^i\mathtt{R}^i[\mathtt{I}| - \mathbf{t}^i]$. If some reprojection error is larger than a threshold, θ, (20pxl), remove all pairwise reconstructions in which some error is larger than θ and go to step 3 (section 6.8.5).

5. **Triangulation**. Triangulate all points in the measurement matrix, $\mathbf{x}$, using depths (section 4.5). A point, p, visible in k views, I_p, $|I_p| = k$, is denoted as an *inlier* if

$$\sum_{i \in I_p} \left(\mathbf{x}_p^i - \Pi(\mathtt{P}^i, \mathbf{X}_p)\right)^2 < \theta_k, \tag{5.2}$$

   where Π is the image projection of a 3D point, $\theta_k = \text{chi2inv}(\gamma, \tau)\sigma^2$ with confidence $\gamma = 0.95$, codimension $\tau = 1$ for $k = 2$ and $\tau = k$ for $k \geq 3$ [31, $p119$], chi2inv is a MATLAB function, and the expected standard deviation of Gaussian noise $\sigma = 0.3$.

6. **Final Robust Bundle Adjustment** with local optimization (section 5.4.2).

Algorithm 2: Multiview reconstruction by robust rotation and translation estimation [67] (section 6.8). The best results can be seen in figures 6.30, $p107$ (high surface quality) and 6.32, $p109$ (2126 images).

the consecutive images often have an overlap. The loops (if any are present) are formed while capturing such part of the scene that has been already captured before. It turns out that these "comebacks" are not much frequent, depending of course on the way of capturing. However, even if there are many comebacks, it is possible to get a reasonable 3D reconstruction from just a few comebacks instead of all of them, whose finding is possible only by matching all $\binom{m}{2}$ image pairs.

There are some alternative approaches avoiding matching all image pairs.

1. In methods [101, 129, 87, 15], descriptors of the features found in all images are placed into a high dimensional space and the clusters found using K-means clustering are used to identify overlapping image pairs. Fast retrieval is achieved using search trees.

   K-means clustering seems to be promising for large datasets [87, 15] (5K images were processed on 30 computers in two days. The expected time for processing 300 images on one computer is about two days.[1]).

2. Methods [56, 80] use a fast tree structure to store and search for image features. In each image, features are detected and stored in the tree. Then, for each feature in each image, the closest (the most similar) features are found in the tree. Besides the feature itself, the result may also contain similar features from the same and other images. By counting the latter cases, one can identify likely overlapping image pairs.

   The same tree can be used to match an image pair. On some data, technique [80] was shown to give comparable results with the exhaustive search (it found about 95% of the tentative matches from the exhaustive search)[2]. In the exhaustive search, all features in the first image are correlated with all features in the other image. Thus, it has quadratic complexity. In practice, it is the most time consuming operation. Therefore, the tree technique is very attractive.

   Some promising experiments with approach [80] were done on matching many images. However, further work is needed to handle issues such as equalization of the tree (what images should be used to build the tree[3]) and the large amount of images.

Although the sub-quadratic complexity makes the above named approaches attractive, due to the fact that they provide only a sub-optimal solution it makes sense to reason about some ways of reducing the number of matched pairs in the framework of pairwise image matching. This section introduces a new algorithm for identifying a subset of EGs needed to join a set of images into one scene via the found EGs. Such subset of EGs can provide a rough 3D reconstruction of the scene. Further, additional EGs may be added to substantially enhance quality of the 3D reconstruction while still using only a small subset of all $\binom{m}{2}$ image pairs.

The New Method

The proposed method exploits a *graph of found EGs*, G. In $G = (V, E)$, vertices V stand for images and an edge in E connects two vertices whenever an EG between the corresponding images with a sufficient support has been found. The set of edges is empty at the beginning and it is updated whenever some EG is found.

The proposed method works in four steps: (i) An initial set of EGs is found such that all views are incident with at least one EG. (ii) Rigid components are identified and further EGs are searched for so that the components are merged, possibly into only one rigid part. Suitable

[1] from personal communication with Ondřej Chum
[2] from personal communication with Štěpán Obdržálek
[3] It is a very time consuming operation (overnight for a few images), which is infeasible for thousands of images.

EGs are located close to articulations. (iii) The "global" spatial connections are strengthened using EGs which shorten long paths in the EG graph. (iv) The "local" spatial connections are strengthened. The detailed explanation of the individual phases follows.

1. **Initialization.** First, an initial set of EGs is found such that all views are incident with some EG. As some sequential structure can be expected for the reasons mentioned at the beginning of section 5.1, it makes sense to prefer the EGs between consecutive views to EGs between views in distance two, etc. After all views become associated with some EG(s), the scene can be partitioned into rigid parts via EGs.

2. **Merging rigid components.** A set of EGs providing a metric 3D reconstruction unique up to an overall similarity transformation will be called a *(rigid) component*. The (rigid) component is two-connected but it is more than that. It can be defined by induction in the following manner. If all three EGs are defined among some image triplet, all images in the triplet belong to the same component.[4] A practical way is to assign each EG a different component at the beginning. Then, for each EG, the EG is merged with (EGs in) all triangles containing it.

   In order to merge rigid components, some EGs have to be found between distinct rigid components. However, there may be many such EGs, thus an additional criterion is needed so that the EGs more likely to exist are tried first. When such an EG is found, some components merge. When this happens, some EGs formerly between distinct rigid components end up within one component. Thus, they are withdrawn from further processing, which means a speedup.

   It has been observed that the more likely EGs are those close to articulations. The classical definition says that an *articulation* is a vertex lying in two distinct two-connected components. In other words, it is an intersection of the two two-connected components. To guarantee uniqueness of estimation of camera translations, we use the stronger definition of the (rigid) component given above.

   After adding new EGs, most articulations disappear while merging components. Now, each of the few rigid component can be reconstructed independently. However, it turned out that it is worth to add some more EGs to significantly improve quality of the 3D reconstruction.

3. **Global strengthening.** The most important EGs are found using the *shortest paths* in a similar way as in [66] (see algorithm 5, *p*93) and strengthened by matching images from their neighborhood (see lower). When using hundreds of images, it would be quite costly to find all shortest paths between all image pairs as in [66]. Instead, an approximation is done by using only one shortest path between the two images found by Floyd-Warshall's algorithm [103] with low complexity $O(m^3)$.

   Not all image pairs from the neighborhood are used. The image pairs are tried only if the shortest path between the two images goes via an important EG. Moreover, only 90% of the most important EGs are strengthened. The reason is that otherwise, gradually, all $\binom{m}{2}$ EGs would be tried.

[4] To be precise, a (rigid) component defined this way suffers an imperfection, which is however small in practice. There may be a whole class of 3D reconstructions when camera centers lie on a line and no three-view correspondences are used [66]. It seems that such a situation can be checked only by estimating the multiview reconstruction. If it turns out that the cameras are colinear, if no three-view correspondences are available to solve for the ambiguity, further EGs may be searched for and the reconstruction reestimated.

4. **Local strengthening.** Finally, the EGs lying in a short distance, e.g. less than 5, are matched. Here, the distance is the length of the shortest path between the two views via known EGs.

In the proposed heuristics, several types of distances and implied neighborhoods are used. Their list follows.

1. Distance between image indices.
2. Distance between image indices in the neighborhood of articulations. The (upper) limit on this distance is used to be higher than the limit on the distance of type 1. The reason is that the latter has a global effect and thus it has to be more limited while the promising (local) neighborhood in the proximity of articulations deserves more attention.
3. Distance to the closest articulation via known EGs.
4. Distance to the closest articulation via (unknown) consecutive EGs, i.e. pairs 1-2, 2-3, etc.

When choosing the most promising EGs, the minimum distance among the named ones is used. The actual setting of the distance limits are up to the user of the heuristics.

Region Types Are Tried Gradually

It turned out that it is possible to decide quite reliably if an image pair has an overlap using one type of distinguished regions only. In our current implementation, if less than 10 tentative matches are found using MSER+ intensity regions, matching of the image pair is given up.

If at least $\frac{s}{\epsilon}$ tentative matches are found, RANSAC on EG is run. Here, s is the prescribed minimum support (number of inliers) for a reliable EG and ϵ is the expected fraction of inliers ($\epsilon = 0.9$). If at least s inliers are found in RANSAC, matching is successfully finished. Otherwise, further detectors are tried gradually.

In our current implementation, Local Affine Frames [70] are constructed on the following regions in the following order: MSER+, MSER- intensity regions, MSER+, MSER- saturation regions, LaplaceAffine and HessianAffine [75] interest points.

If a reliable EG was found and additional matches are needed, they are found using SIFT features [56] constructed on the same regions and on Lowe keypoints [56]. It was observed that Lowe features with SIFT descriptors are very unreliable on some difficult scenes such as the Tête scene [67] (see figure 6.28, *p*106) where many correspondences were built using these features on rocks between images with no overlap. A possible reason is that the Lowe keypoints do not possess affinely covariant frames (and thus descriptions) such as Local Affine Frames.

5.2 Joining Matches into Tracks

It is supposed that pairwise EGs are provided by some correspondence matcher such as described in the previous section. The quality of the found EGs may differ substantially. Some EGs may be only slightly corrupted by a few mismatches[5] while some EGs may be entirely wrong like in the case of the so-called non-existent EGs [67] (see figure 6.22, *p*99).

In the original method [63] all inliers w.r.t. the EG were placed into the measurement matrix (MM) while simply ignoring conflicting matches. Many outliers were then removed in a subsequent stage using trifocal tensors [61]. However, this turned out not to work when there were many image pairs with no overlap.

[5] Not talking about (i) the always present noise in point positions caused by sampling and quantization of (noisy) optical signal and (ii) uncorrected defects on camera optics such as the radial distortion.

Suppose the object has two or more parts looking same. To reduce the risk that too many matches between similar regions on different parts of the object would result in a wrong reconstruction of the cameras, it might seem necessary to forbid matching between images seeing these different parts. However, it is possible to phase-out most of such image pairs automatically using EGs only, as described below. There will always be some matches between image pairs with no mutual overlap and (incorrect) EGs will be estimated with usually just a few matches satisfying them. Still, the number of matches may be higher than in some other image pair with a correct EG but with only a few matches due to small image overlap. Therefore, discarding pairs with the number of matches falling below a threshold cannot work.

A simple greedy algorithm can overcome this difficulty quite satisfactorily: First, matches from the image pair with the most inliers with respect to the EG are loaded into the MM. Next, matches from the pair with the second largest number of inliers are loaded etc. This guarantees that more reliable matches are used first. Each match is checked against so-called *reliable EGs* taken as those with the number of inliers above some threshold (30 in default). The inliers are counted as the original inlying matches from the image pair plus new matches added into the MM from other pairs. If a match satisfies all reliable EGs, it is merged into the MM. It turns out that many outliers and only a few inliers are discarded this way.

In our current implementation [67] (section 6.8), the matches not satisfying some reliable EGs are simply added as new tracks into the MM. The reason is that they may be correct but were recognized as incorrect due to some wrong EG incorrectly recognized as reliable.

For a very large data with thousands of high-resolution images, a special attention has to be given to the data structures for storing the MM due to both memory and time issues. MATLAB's sparse matrix is too slow for the operation of adding new non-zero elements especially when the matrix has already a huge number of elements. Therefore, we implemented our own sparse matrix using a hash table in the C language[6].

5.3 Focal Length Estimation

In [66], the overall scale of the focal length was estimated as the mean of the six-point [111] estimates from all image pairs weighted by the square of the EG support (see *p*88).

After trying various ways, the following method proved to work well on most scenes. In our current implementation, we estimate the focal length as

$$f \quad = \quad \exp\left(\frac{1}{n}\sum_{i=1}^{n}\log(f_i)s_i{}^2\right), \tag{5.3}$$

where f_i and s_i are the focal length estimates and the numbers of inliers, respectively. Similarly as projective depths [65] (section 6.3.6), the focal length "lives" on the half line $(0, \infty)$. It seems that the error in the focal length estimation has a normal distribution but in the logarithmic scale. Note that the result of (5.3) is independent of rescaling the focal length and the logarithm base.

We observed that the log-scale variant (5.3) gives better estimates of the focal length which proves by that the final BA (with varying focal length) changes the focal length less. Also, more image triplets were constructed from the found EGs (using the five-point algorithm) on, e.g., the Raglan scene [67] (see figure 6.26, *p*105).

As the focal length estimation using the six-point algorithm [111] is unstable for points lying on or close to a plane, estimates from such EGs should not be used in the focal length calibration (5.3). Such EGs either (i) have all points on or close to a plane or (ii) form a panorama,

[6]The implementation was done by Jan Toušek from Neovision s.r.o., Czech Republic.

which proves in a small baseline angle (see section 5.5, *p*37). However, it may happen that the baseline angle gets large due to bad focal length estimation. Such situation may be difficult to detect because the dominant plane is not guaranteed to be found (see section 4.3, *p*22). Therefore, more samples (30% of the widest-baseline EGs) should be used in equation (5.3) to gain robustness w.r.t. such situations. Another possibility is to use median on focal length estimates (without using the log-scale).

5.4 Bundle Adjustment

An overview of the non-linear Levenberg-Marquardt optimization called bundle adjustment (BA) can be found, e.g., in [76, 31, 54].

5.4.1 Projective Bundle Adjustment

Projective BA was used only after projective factorization in publications till [62] (section 6.1). In projective BA, perspective cameras are parameterized as an eleven-vector (the 3×4 camera matrix determines the camera up to a non-zero scale).

Before projective BA, a quasi-affine upgrade [131, 31] is made from the projective reconstruction to get all cameras to the same side of the plane at infinity. This is equivalent to that all cameras have the same sign of determinant of the first three columns of the projection matrix, $\det \mathtt{P}(:, 1:3)$ (either all positive or all negative), which also means that all cameras have the same handedness.

It has been observed that when some cameras have positive and some negative determinants, it is not possible for the BA to change the determinant sign in projective camera parameters. Thus, in such a case, the BA ends up in a much worse local minimum than it could. We do not know why this problem appears as the used parameterization enables (even for affine cameras) the determinant to be zero. Whereas, in the transformed space of the quasi-affine upgrade, the determinants of all cameras are all positive, for instance, and their varying during the BA amounts no numerical difficulties.

5.4.2 Metric Bundle Adjustment

Our C-implementation of the metric BA was based on publicly available software [54]. Many improvements were added such as

- usage of various camera models (perspective, omnidirectional, several types of the radial distortion),
- various types of rotation parameterization (4-quaternions, incremental rotations [40]).
- The secondary sparsity of the Hessian is exploited. It is simply implemented by a call to MATLAB and building a MATLAB's sparse matrix. This enables to solve much larger problems as no memory space is allocated for zeros in the Hessian and it is also much faster. Tens of thousands of cameras and millions of points are no problem with a few GB of memory.
- When desired, the oriented projective geometry *(OPG)* [131] is checked for each point (if it lies in front of all cameras it is observed in) after each step of the BA. If some point harms the OPG, the optimization is interrupted as such point would harm the optimization. Then, such point is either (i) discarded from further BA steps, or (ii) it is triangulated so that OPG is satisfied [44] before the next step of the BA.

- The code was further optimized by exploiting, e.g., the fact that most of the used matrices [54] are symmetric.

  Perhaps the best publication on fast BA implementation is [18].

Robustness w.r.t. Mismatches

The following techniques were implanted into the BA routine with the aim of making BA more robust to mismatches and inaccuracies in the initial estimate of cameras and 3D points.

Let the residual of an image point, $\mathbf{r}$, be defined as $\mathbf{r} = \mathbf{p} - \mathbf{x}$ where $\mathbf{x}$ and $\mathbf{p}$ are the measured and reprojected image point, respectively. The techniques follow.

1. **Robust BA using Weights** computed using the Gaussian kernel as

   $$\omega \quad = \quad e^{-\frac{s}{\sigma^2}}$$

   where $s = \sum_{c=1}^{C} {\mathbf{r}_c}^2$ is the sum of squares of the image point residual along its C coordinates[7] and the σ parameter of the Gaussian controls how large residuals have significant influence. Note that this is similar but much more simple than MLESAC [123].

2. **Robust Kernel in BA**. In the BA, the Lorentzian robust kernel [20, section 5]

   $$\epsilon_1(\mathbf{r}) \quad = \quad \log(1 + \frac{|\mathbf{r}|}{\sigma})$$

   is used as well as its following variation:[8]

   $$\epsilon_2(\mathbf{r}) \quad = \quad \mathbf{x} + \text{sign}(\mathbf{r})\sigma \log(1 + \frac{\mathbf{r}^2}{\sigma^2}).$$

   The sign is needed here to ensure that the reprojection $\mathbf{p}$ corrected by the kernel appears on the correct side of the measured point $\mathbf{x}$. Both variations give similar results. However, the latter is preferred as it suppresses the influence of the points with high residuals more.

3. **Local Optimization**. Bundle adjustment is iterated using a threshold on precision which decreases with each iteration. Points with reprojection errors above the precision are not used in the optimization. This technique is called *local optimization* [13].

Finally, methods 1 and 2 were not used in any publication as better techniques for handling mismatches and detecting non-existent EGs were developed in [67] (section 6.8).

5.5 Detection of Image Pairs with Very Short Baseline

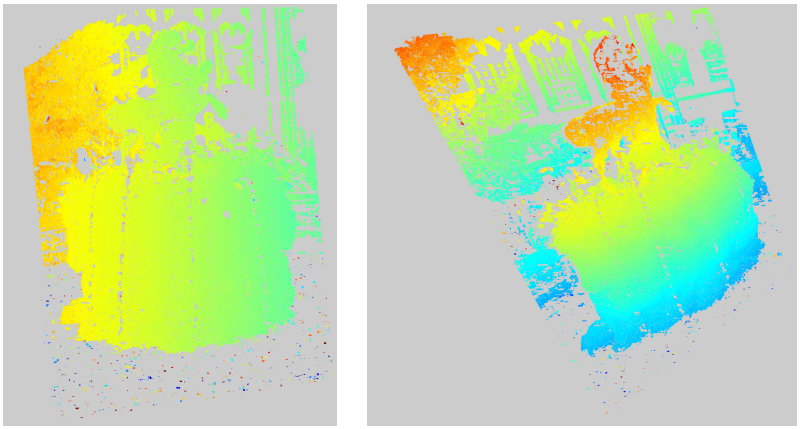
In the pipeline (algorithm 1, *p*29, [17]), after obtaining cameras using the multiview reconstruction (see chapter 6), dense stereo is run on some image pairs to obtain a dense reconstruction. It is desired to forbid all pairs not suitable for dense stereo. These are especially pairs with (nearly) coinciding camera centers (these form a panorama). Two methods were developed.

[7] C equals two or three for the perspective or the omnidirectional camera, respectively

[8] One could try to apply the square root on the logarithm so that $\mathbf{r}$ is not squared twice (first inside the logarithm and second in the Levenberg-Marquardt routine):

$$\epsilon_4(\mathbf{r}) \quad = \quad \sqrt{\log(1 + \frac{\mathbf{r}^2}{\sigma^2})}$$

However, when $\mathbf{r}$ is zero, $\log(1) = 0$ and consequently the derivative of the above function becomes $log(0)^{-\frac{1}{2}} \cdot \ldots$ and thus produces a NaN (Not a Number) due to the division by zero. Thus, sqrt is not applied here and one minimizes squares of logarithms in BA.



21.41% outliers 0.30% outliers

Figure 5.1: Detection of very short baseline in the Detenice Fountain scene [2]: image pair with very short baseline produces a false disparity map (left) compared to a correct disparity map of a pair with non-negligible parallax (right). Under each image, amount of outliers w.r.t. the panorama model within accuracy $\sigma = 4\sigma_0$ is shown.

- **Fitting the Panorama Model** [67] (section 6.8.6). If some pair should fit a panorama model, i.e. camera rotation and no translation, it must fit a weaker homography model at least so well. Thus, only pairs with 90% inliers lying on a (dominant) plane need to be checked for being a panorama, the remaining ones cannot be a panorama.

  In our implementation, (dominant) planes are detected during DEGENSAC [16] on image pairs, see section 4.3. However, it happens (although rarely) that no H-degenerate sample is drawn even when some are present in the data due to that sampling ended up too early. In other words, DEGENSAC [16] does not guarantee that the plane is found provided that it is present in the data. To correctly determine the presence of a plane or a panorama, one should apply either RANSAC on a plane or the following test on the panorama model.

  Fitting the panorama model was started by unifying the two camera centers by setting them to their mean. Then a BA constrained to keep the camera centers same was run. Finally, local optimization [13] using the same BA was iterated while decreasing the threshold on precision [13, section 3] with each iteration. The threshold was initialized to double of the mean reprojection error resulting from the first BA followed after unifying the camera centers.

  It turned out that it is better to relax the precision, σ, (the used factor is four, $\sigma = 4\sigma_0$, where the expected standard deviation of Gaussian noise in measurements, σ_0, is set in default to $\sigma_0 = 0.3$, as in [13, section 4]) and to require that 95% of all points fit the panorama model. The threshold on the sum of squares of the reprojection errors of a 3D point is $\theta = \text{chi2inv}(\gamma, \tau)\sigma^2$ where confidence was set to $\gamma = 0.95$, codimension $\tau = 1$ for two views [31], and chi2inv is a MATLAB function. See figure 5.1.

- **Baseline Angle**. Recently, a much simpler way of detecting very short baselines was found. For each ij-EG, *baseline angle*, α^{ij}, between one camera center, the point cloud center, and the other camera center is computed. More precisely,

$$\alpha^{ij} \quad = \quad cos^{-1}\left(\frac{\mathbf{v}^{i\top}\mathbf{v}^{j}}{||\mathbf{v}^{i}||_2||\mathbf{v}^{j}||_2}\right) \tag{5.4}$$

where $\mathbf{v}^i = \mathbf{C}^i - \mathbf{X}_{cen}$ is the vector between the center of the 3D point mass, $\mathbf{X}_{cen}$, and the i^{th} camera center, $\mathbf{C}^i$, in the ij-reconstruction and $||\cdot||_2$ is vector 2-norm.

Only image pairs with the baseline angle above some threshold, $\alpha^{ij} > \alpha_{min}$, are used for dense stereo. In default, α_{min} is set to 3 degrees.

6 Multiview Reconstruction Estimation

In this chapter, several methods for 3D multiview reconstruction are presented. All presented methods can deal with the perspective camera model and occlusions. Only sections 6.3.4 and 6.4 are devoted to more simple and weaker affine camera model. Some methods can handle mismatches, the newest ones can handle also dominant planes, panoramas and non-existent epipolar geometries. The history of the methods can be seen in table 6.1, *p*42.

A brief description of the methods follows (in chronological order).

1. **Projective Factorization** (section 6.1) [62] handles the perspective camera model (projective depths are estimated using method of Sturm and Triggs [113]) while occlusions are handled by Jacobs [43]. 3D reconstruction via projective factorization is performed in the following steps:
   a) Cameras (i.e. the *column basis* of the MM, see *p*7) are found by applying constraints on them formed from three- or four-tuples of columns of the MM.
   b) Optionally, the reconstruction can be improved by filling of the MM using the projected points and *factorizing* it using SVD (with possible iterations) and using bundle adjustment.
      However, during filling the MM, new errors are introduced into the MM originating from the errors in the camera estimates. Therefore, SVD on the filled MM is biased by the camera estimates and thus does not necessarily have to provide a better estimate than step 1a.
   c) Finally, a *metric upgrade* [31] from the projective to the metric space is performed, usually necessarily followed by bundle adjustment.

   In this process, step 1a is crucial because once the cameras are found, the following steps can only improve the solution. However, they can be reasonably used only when the data is substantially cleared of outliers.

   A more serious problem with the projective factorization is that it does not provide sufficiently accurate result for the metric upgrade (step 1c) on such simple scenes like the Dinosaur sequence (figure 6.9, *p*67). The reason is that Jacobs [43] uses complementary subspaces but these do not correctly represent image measurement error when noise is present. A more correct solution in the original subspaces was proposed in [65] (section 6.3).

2. **Mismatch Detection by Trifocal Tensor Voting** (section 6.2) [61]. Although a substantial amount of mismatches can be rejected even before they are placed into the MM (section 5.2), some mismatches remain in the MM. In [61], an image point in the MM was confirmed to be an inlier if it gathered enough votes from trifocal tensors sampled on six-tuples of points seen in image triplets [102].

   In the newer papers, validation using trifocal tensors was not used as better methods for handling mismatches were developed like, e.g., [67] (section 6.8).

3. **Projective Gluing** (section 6.3) [65]. Jacobs' handling of occlusions [43] was reformulated in the original subspaces, gaining so a stable solution even for hundreds of images.

Consistent projective depths were estimated using an overdetermined system of equations, which combined depth estimates from many EGs, gaining so more accurate depth estimates and consequently more accurate reconstructions.

This method represents the most significant achievement in history of all the developed methods. All latter ones build on it.

Section 6.4 revisits gluing via affine cameras.

4. **Projective Gluing without Depth Consistency** (section 6.5). Unlike [65] (section 6.3), no three-view correspondences (and depths – hence the method name) are needed. The scale consistency between projection matrices (obtainable from the EGs) is obtained by solving a system of equations in which some scales are estimated besides the camera matrices. The basic form of the technique for three views can be used to estimate three-view reconstructions given EGs only, which can be further combined to provide a multiview reconstruction.

   Compared to [65] (section 6.3), this method has quite low accuracy of the 3D reconstruction resulting from not using the three- and more-view correspondences.

   In the latter methods, camera calibration was exploited which increased the precision significantly. Rotations and translations were solved separately, the scale consistency was no longer an issue as it was part of the translation registration. Therefore, this method (section 6.5) has never been published.

5. **Merging Panoramas**, or "A Successful Approach for the ICCV'05 Contest" (section 6.6). A new method for estimating the focal length and a homography aligning two images in a panorama was developed. A variation of projective gluing [65] (section 6.3) was used to align all images in a panorama given pairwise alignments. The built panoramas were used for 3D reconstruction of the scene by estimating reconstructions of view triplets and merging these. A technique for 3D reconstruction of view triplets by making camera rotations consistent was presented. Camera translations were estimated afterwards so that points got in front of cameras using the cheirality constraint [31].

   The translation estimation was further in [66] (section 6.7) replaced by the L_∞-minimization [44]. Thanks to that, panoramas do not have to be detected in advance any more (which is a difficult problem).

6. **Metric Gluing**, or "How to Achieve a Good Reconstruction from Bad Images" [66] (section 6.7). Pair-wise metric reconstructions given up to similarities are glued by (i) registering camera rotations linearly, (ii) refining the pair-wise reconstructions while keeping the rotations consistent using bundle adjustment, and (iii) estimating camera translations using Second Order Cone Programming by minimizing the L_∞-norm [44]. Unequiponderantly captured images are better handled using weights resulting from a criterion for evaluating importance of an epipolar geometry in influence on the overall 3D geometry.

7. **Robust Rotation and Translation Estimation** [67] (section 6.8). Compared to [66] (section 6.7), rotation registration was improved by simplifying the equations and thus enhancing the stability. For translation estimation, only four points per EG are used, gaining so a significant speedup. Robustness is achieved by using a criterion that diminishes the risk of choosing a mismatch. Non-existent EGs found in scenes with repetitive and similar structures are detected as the ones with the largest residual and removed. Iterative reregistration removes most non-existent EGs.

All the above methods can be used for omnidirectional cameras as well [73].

Finally, we give an example which demonstrates differences among the above methods.[1] Suppose one has partial metric (two-, three- or more-view) reconstructions at hand and wants to obtain a multiview reconstruction using some of these methods. First, the partial reconstructions can be glued projectively (even though they are more than projective). Methods in sections 6.3 and 6.5 require consistent and inconsistent scalings of cameras, respectively. The former will give more accurate results if applicable, i.e. if three-view correspondences are available. Another option is to first register rotations and then translations (sections 6.7 or 6.8). Although all the named methods can be used in principle, the last one will most likely give the most stable, robust, and accurate result.

Table 6.1 demonstrates that newer methods are also more capable.

Radial Distortion

The radial distortion was removed in the final metric bundle adjustment only in work [17]. In newer papers, radial distortion was not removed from two reasons: first, it seemed that the distortion model overfited. To prevent that, more points would be needed. One could use millions of points from dense stereo. Secondly, in our experiments, the simple perspective camera model provided high quality reconstructions even without compensation of the radial distortion (see sections starting from 6.3).

[1] One can use similar methods for reconstruction from lines, see section 7.2.

Table 6.1: History of the developed multiview reconstruction methods for points, their novelty and applicability. Methods are enumerated in chronological order. Each method can reconstruct all scenes shown above it (except [73]) and, vice versa, no method is able to reconstruct any scene shown below it.

Method	Section	Work	Novelty	Applicability
projective factorization	6.1	[62]	perspective camera & occlusions	Temple[a]
	6.2	[61]	outlier detection by trifocal tensor voting	Castle
	5.2	[63]	fusion with correspondence estimator [69]	Valbonne
		[17]	fusion with dense stereo [52]	Head1
		[73]	omni-directional cameras [74]	Venice
projective gluing	6.3	[65]	[43] in original subspaces, consistent depths	Dinosaur
gluing without depths	6.5		projective gluing without depths	
merging panoramas	6.6	[59]	panorama detection, rotation registration[b], translation registration for two and three views	ICCVC'05
metric gluing	6.7	[66]	rotation registration, translation registration [44]	Head2
robust rotation and translation estimation	6.8	[67]	registering relative rotations, robust translation registration, non-existent EG removal	Raglan

[a] Reconstruction of the Dinosaur sequence published in [62] is wrong, which turned out after the metric upgrade done in [65] (section 6.3.1).

[b] this was published later in [66] (section 6.7)

6.1 Factorization with Perspective Cameras and Occlusions

This section[2] presents a method for recovery of projective shape and motion from multiple images by factorization of a matrix containing the images of all scene points. This method can handle perspective views and occlusions jointly. The projective depths of image points are estimated by the method of Sturm & Triggs [113] using epipolar geometry. Occlusions are solved by the extension of the method by Jacobs [43] for filling of missing data. This extension can exploit the geometry of perspective camera so that both points with known and unknown projective depths are used. Many ways of combining the two methods exist, and therefore several of them have been examined and the one with the best results is presented.

A complete rescaled measurement matrix has rank four and therefore a projective reconstruction can be obtained by its factorization. However, from measurements in perspective images with occlusions, we can only compose a measurement matrix which is neither complete nor rescaled. When it is at all possible to compute projective depths of some known points in PRMM `R` (see equation (2.2), *p*7), e.g. via multi-view constraints, some missing elements of `R` can often be filled using the knowledge that every five columns of complete rescaled `R` are linearly dependent.

It would be ideal to first compute the projective depths of all known points in `R` and then to fill all the missing elements of `R` by finding a complete matrix of rank four that would be equal (or as close as possible) to the rescaled `R` in all elements where `R` is known. Such a two-step algorithm is almost the ideal linearized reconstruction algorithm, which uses all data and has a good statistical behavior. We have found that many image sets, in particular those resulting from wide baseline stereo, can be reconstructed in such two steps.

Of course, there are image sets, e.g. sets with the structure of missing data on the borderline of reconstructibility or long sequences with very fractionalized tracks, which cannot be solved in the above two steps. Instead, the two steps have to be repeated while the measurement matrix `R` is not complete. If the correspondences between the images are such that the measurement matrix is large and diagonally dominant, then it is possible to use another reconstruction technique, e.g. to fuse partial consecutive reconstructions [21, 7]. However, if there is no clear sequence of images or central image like in [127], the presented algorithm has a clear advantage. It can handle arbitrary scenes in pseudo-optimal manner without a priori preferring any particular image. It provides a unique solution and thus is suited for the initialization of bundle adjustment optimizations.

In what follows, we shall describe the two steps of the algorithm. Let us first review the two steps we build on and their respective extensions. Later we will describe how to combine the two steps.

6.1.1 Estimating the Projective Depths

Many works dealt with estimating the projective depths. In this work, we used Sturm & Triggs' method [113] exploiting epipolar geometry but other methods, e.g. [35, 58, 30], can be applied too. Method [113] was proposed in two alternatives. The alternative with a central image is more appropriate for wide baseline stereo while the alternative with a sequence is more appropriate for video-sequences. The former will be denoted as $\omega_{cent,c}$ where c denotes the index of the central image while the latter will be denoted as ω_{seq}. Thus, we have altogether the totality $\Omega = \{\omega_{seq}, \omega_{cent,1} \dots \omega_{cent,m}\}$ of alternatives for computing the projective depths. Also, the method from [113] has to be furthermore slightly modified on account of missing data. The

[2] Most of this section was published in [62]. Marc Pollefeys from K.U.Leuven provided the Temple data and Tomáš Werner from the University of Oxford provided the routine for the bundle adjustment.

1. Set $\lambda_p^j = 1$ for all p corresponding to known points $\mathbf{x}_p^j$ in view $j = \begin{cases} 1: & \text{for } \omega_{seq} \\ c: & \text{for } \omega_{cent,c} \end{cases}$

2. For $\begin{cases} j = 1 \dots m-1, & i = j+1: & \text{for } \omega_{seq} \\ j = c, & i \neq j \quad : & \text{for } \omega_{cent,c} \end{cases}$ do the following. If images i and j have enough points in common to compute a fundamental matrix uniquely[a] then compute their fundamental matrix $\mathtt{F}^{ij}$, epipole $\mathbf{e}^{ij}$, and depths λ_p^i according to

$$\lambda_p^i = \frac{(\mathbf{e}^{ij} \wedge \mathbf{x}_p^i) \cdot (\mathtt{F}^{ij}\mathbf{x}_p^j)}{\|\mathbf{e}^{ij} \wedge \mathbf{x}_p^i\|^2} \lambda_p^j$$

if the right side of the equation is defined, where $\wedge$ stands for the cross-product.

For ω_{seq}: if the p^{th} track[b] ($p = 1 \dots n$) is discontinuous, start with $j = b(p)$ where $b(p)$ denotes the initial image of the longest continuous subtrack of the p^{th} track.

[a] See section 6.1.4.
[b] The p^{th} track denotes a subsequence of known points in sequence $\mathbf{x}_p^1 \dots \mathbf{x}_p^m$.

Algorithm 3: Estimating the depths: alternatives ω_{seq} and $\omega_{cent,c}$.

complete algorithm is summarized in algorithm 3. The geometrical explanation is provided in figure 4.1, *p*19.

6.1.2 Filling of Missing Elements in R

Filling of missing data was first realized by Tomasi & Kanade [119] for orthographic camera. D. Jacobs [43] improved their method and we use our extension of his method for the perspective case. Often, not all depths can be computed using alternatives Ω because of the missing data.[3] Therefore, we extend the method from [43] so that also points with unknown depths are exploited. Moreover, the extension is independent of how depths are estimated and thus any method for estimating the depths could be used. Before describing our modification for the perspective camera, the original Jacobs' algorithm for the orthographic case has to be explained.

D. Jacobs treated the problem of missing elements in a matrix as fitting an unknown matrix of a certain rank to an incomplete noisy matrix resulting from measurements in images. Assume noiseless measurements for a while to make the explanation simpler. Assuming perspective images, an unknown complete $3m \times n$ matrix $\tilde{\mathtt{R}}$ of rank four is fitted to PRMM $\mathtt{R}$ (see equation (2.2), *p*7). Technically, a basis of the linear vector space that is spanned by the columns of $\tilde{\mathtt{R}}$ is searched for. Thus, when there are four complete linearly independent columns in $\mathtt{R}$, then they form the desired basis. When no such quadruplet of columns exists, the basis has to be constructed from incomplete columns. Fortunately, some quadruplets of incomplete columns provide constraints on the basis and a sufficient number of such constraints determine it.

Let us explain what we mean by saying that an incomplete column $\mathbf{c}$ of $\mathtt{R}$ spans (generates) a subspace. Every complete column of $\mathtt{R}$ generates a one-dimensional subspace of $\mathbb{R}^{3m}$. Thus, an incomplete $\mathbf{c}$ generates a subspace V, as the smallest linear space containing all one-dimensional subspaces generated by $\mathbf{c}$ after replacing unknown elements by some arbitrary real numbers. Linear subspaces form a complete lattice [8] and therefore such smallest linear space V exists. It is a subspace of $\mathbb{R}^{3m}$ and equals the linear hull of all one-dimensional subspaces. The generators

[3] In a later publication [65] (section 6.3), it turned out that all depths can be estimated using a non-minimal set of epipolar geometries. The bilinear problem of inconsistent scalings of the $\mathtt{F}^{ij}$ matrices is transformed into a linear problem by applying logarithm on positive depths.

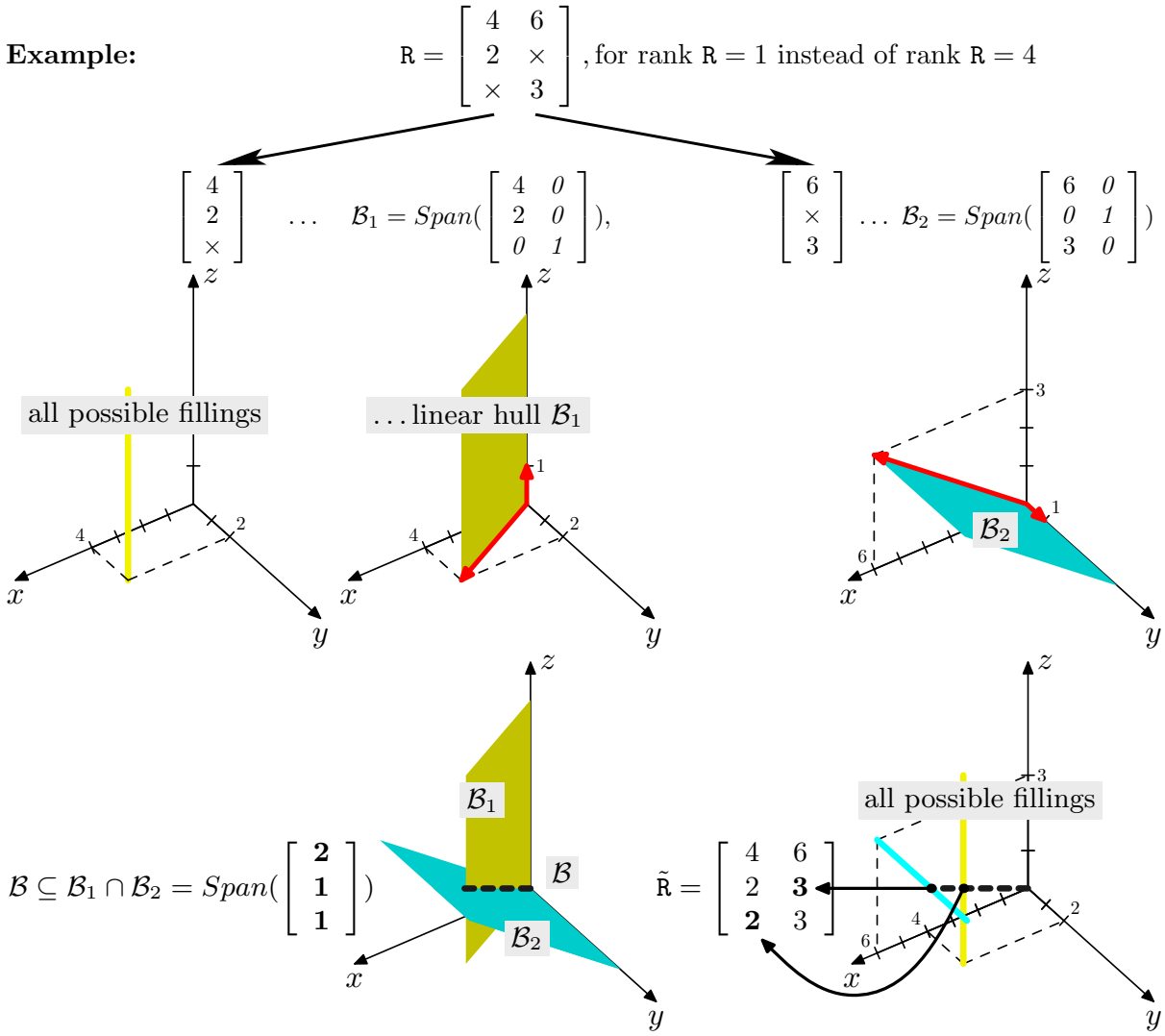


Figure 6.1: Forming constraints on the basis and filling the matrix. For $\mathtt{R}$ is of rank one, constraints on $\mathcal{B}$ are formed by single columns.

of V can be obtained by constructing the column containing the known elements of $\mathbf{c}$ and zeros instead of the unknown ones and augmenting it with the standard basis spanning the dimensions of the unknown elements (see figure 6.1 and the example in section 6.1.3).

Let the space generated by the columns of $\tilde{\mathtt{R}}$ be denoted by $\mathcal{B}$. Let $\mathcal{B}_t$ denotes the span of the t^{th} quadruplet of columns of $\mathtt{R}$ which are linearly independent in coordinates known in all four columns. $\mathcal{B}$ is included in each $\mathcal{B}_t$ and thus also in their intersection, i.e. $\mathcal{B} \subseteq \bigcap_{t \in T} \mathcal{B}_t$, where T is some set of indices. When the intersection is 4D, $\mathcal{B}$ is known exactly. If it is of a higher dimension, only an upper bound on $\mathcal{B}$ is known and more constraints from quadruplets must be added. Any column in $\tilde{\mathtt{R}}$ is a linear combination of vectors of a basis of $\tilde{\mathtt{R}}$. Thus, having a basis $\mathtt{B}$ of $\tilde{\mathtt{R}}$, any[4] incomplete column $\mathbf{c}$ in $\mathtt{R}$ can be completed by finding vector $\tilde{\mathbf{c}}$ generated by $\mathtt{B}$ which equals $\mathbf{c}$ in the elements where $\mathbf{c}$ was known in $\mathtt{R}$ (see figure 6.1).

Linear independency of the quadruplet of columns is crucial to obtain a valid constraint on the basis. Consider, e.g., a quadruplet consisting of four equal columns, thus spanning only a 1D

[4] containing at least four known elements, which in practice means six elements resulting from two known points

space. Even if three coordinates in one of its columns are made unknown, and thus a 4D space is spanned, $\mathcal{B}$ does not have to be included in the span. A row with some missing coordinates can be ignored because the entire corresponding dimension is spanned and the constraint on $\mathcal{B}$ is always satisfied in the dimension, meaning such a row contains no information. This is the reason to use just the quadruplets of columns linearly independent in coordinates known in all four columns.

Because of noise in real data, the intersection $\bigcap_{t\in T}\mathcal{B}_t$ quickly becomes empty. This is why $\mathcal{B}$ is searched for as the closest 4D space to spaces $\mathcal{B}_t$ in the sense of the minimal sum of square differences in known elements. Denoting complement of a linear vector space by $\perp$, $\bigcap_{t\in T}\mathcal{B}_t$ can be expressed according to the well known De Morgan rule as $(\mathrm{span}_{t\in T}\,\mathcal{B}_t^\perp)^\perp$. The generators of $\mathcal{B}_t^\perp$ can be found as $\mathtt{B}_t^\perp = \mathtt{u}(:, d+1 : end)$, where $[\mathtt{u},\mathtt{s},\mathtt{v}] = \mathrm{SVD}(\mathtt{B}_t)$ and d is the dimension of $\mathcal{B}_t$. $\mathrm{span}_{t\in T}\,\mathcal{B}_t^\perp$, where T is of cardinality z, is generated by $[\mathtt{B}_1^\perp \mathtt{B}_2^\perp \dots \mathtt{B}_z^\perp]$. $(\mathrm{span}_{t\in T}\,\mathcal{B}_t^\perp)^\perp$ is generated by $\mathtt{u}(:, end-3 : end)$, where $[\mathtt{u},\mathtt{s},\mathtt{v}] = \mathrm{SVD}([\mathtt{B}_1^\perp \mathtt{B}_2^\perp \dots \mathtt{B}_z^\perp])$.

6.1.3 Filling of Missing Elements for the Perspective Camera

Jacobs' method [43] cannot use image points with unknown depths. But, the PRMM constructed from measurements in perspective images often has many such points where the corresponding depths cannot be computed.[5] Therefore, we extended the method to exploit also points with unknown depths. It brings two advantages: (i) because the actual iteration of the two-step algorithm exploits more information, the number of iterations may decrease and consequently more accurate results may be obtained; (ii) it is possible to reconstruct more scene configurations. See [60, section 8] for more details about this. It is important that the presented extension is still a linear method as was the Jacobs' method [43].

Let us first explain the extension for two images. Suppose that λ_p^i and $\mathbf{x}_p^i$ are known for $i = 1, 2$, and for $p = 1 \dots 4$ except λ_4^2. Then, consider the first four columns of $\mathtt{R}$ to be the t^{th} quadruplet of columns, $\mathtt{A}_t$. A new matrix $\mathtt{B}_t$, whose span will be denoted by $\mathcal{B}_t$, can be defined using known elements of $\mathtt{A}_t$ as

$$\mathtt{A}_t = \begin{bmatrix} \lambda_1^1\mathbf{x}_1^1 & \lambda_2^1\mathbf{x}_2^1 & \lambda_3^1\mathbf{x}_3^1 & \lambda_4^1\mathbf{x}_4^1 \\ \lambda_1^2\mathbf{x}_1^2 & \lambda_2^2\mathbf{x}_2^2 & \lambda_3^2\mathbf{x}_3^2 & ?\ \mathbf{x}_4^2 \end{bmatrix} \longrightarrow \mathtt{B}_t = \begin{bmatrix} \lambda_1^1\mathbf{x}_1^1 & \lambda_2^1\mathbf{x}_2^1 & \lambda_3^1\mathbf{x}_3^1 & \lambda_4^1\mathbf{x}_4^1 & 0 \\ \lambda_1^2\mathbf{x}_1^2 & \lambda_2^2\mathbf{x}_2^2 & \lambda_3^2\mathbf{x}_3^2 & 0 & \mathbf{x}_4^2 \end{bmatrix}$$

It can be proved (see corollary 1 in [60, appendix A]) that if $\mathtt{B}_t$ is of full rank (i.e. five here) then $\mathcal{B} \subseteq \mathrm{span}(\mathtt{B}_t)$, which is exactly the constraint on $\mathcal{B}$.

In a general situation there are also some missing elements in $\mathtt{R}$. Then, the matrix $\mathtt{B}_t$ is constructed from the t^{th} quadruplet $\mathtt{A}_t$ of columns of $\mathtt{R}$ as follows:

1. Set $\mathtt{B}_t$ to $\mathtt{A}_t$.
2. Replace all unknown points and points with unknown depth by zero in $\mathtt{B}_t$.
3. For each unknown depth λ_p^i in $\mathtt{A}_t$, add a column with $\mathbf{x}_p^i$ and zeros everywhere else to $\mathtt{B}_t$.
4. For each triplet of rows in $\mathtt{A}_t$ containing some unknown point, add to $\mathtt{B}_t$ the standard basis spanning the dimensions of the unknown point.

An example demonstrating the construction of $\mathtt{B}_t$ from a quadruplet $\mathtt{A}_t$ can be seen in figure 6.2. If $\mathtt{B}_t$ is of full rank, its span $\mathcal{B}_t$ includes $\mathcal{B}$ (this can be proved by induction from corollary 1 in [60, appendix A]). By including also image points with unknown projective depths spaces $\mathcal{B}_t$ spanned by quadruplets of columns become smaller, thus solving the complete problem becomes more efficient.

[5] see footnote 3, *p*44

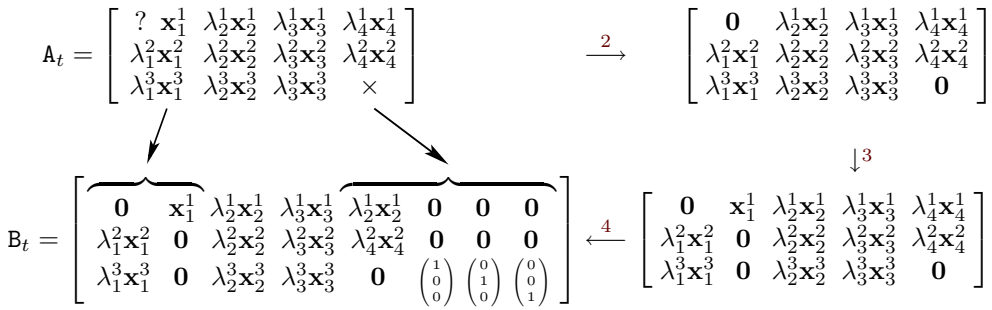


Figure 6.2: An example of construction of $\mathtt{B}_t$ from a quadruplet $\mathtt{A}_t$.

It can be seen that the concept of generating constraints on the basis for the orthographic case is only a special case of generating constraints for the perspective case. The former is equivalent to the latter having all depths set to the same number thus corresponding to the perspective camera with the projection center at infinity and looking at a finite scene.

6.1.4 Combining the Filling Method with Estimating the Depths

Due to occlusions, the computation of projective depths can be carried out in various ways depending on which depths are computed first and if and how those already computed are used to compute the others. One way of depth estimation will be called a *strategy*. Depending on the chosen strategy, different subsets of depths are computed and different submatrices of the PRMM are filled. It may happen when some strategy exploiting, e.g., epipolar geometry of some pair of images is used that the fundamental matrix cannot be computed due to occlusions. Consequently, depths needed to form a constraint on the basis of the PRMM in one of the images cannot be estimated, thus the missing data in the image cannot be filled and the two steps of depth estimation and filling has to be repeated.

For accurate data, all strategies should be equivalent. It is not so if the data is noisy. In such case, the task is to choose the strategy which results in the smallest error. It would be unrealistically costly to compute all possibilities (although there is "only" a finite number of them[6]) and to choose the best one. Fortunately, we do not have to compute all of them in order to find some good one. From the structure of the missing data, it is possible to predict a good strategy for depth estimation that results in a good reconstruction. Some criterion deciding which strategy is good is needed. For scenes reconstructible in more steps, such criterion also determines which subset of depths is better to be computed first.

The following two observations have been made. First, the more iterations are performed, the less accurate results are obtained because the error from the former iteration spreads in subsequent iterations as was also mentioned in [43]. Secondly, unknown elements should not be computed from fewer data when they can be computed from more data, and thus more accurately due to the law of big numbers and the assumption of random noise. Both these observations support the following.

Principle 1 *The more image points that are filled in one step, the smaller the expected error.*

This principle leads to a pseudo-optimal number of iterations that need to be performed.[7]

[6] All strategies correspond to all minimum sets of fundamental matrices, i.e. trees on a (possibly) complete graph. See section 6.3.7.

[7] An optimal strategy would have to be searched for as the shortest branch in the tree graph of all partial solutions. Partial solutions can be ordered into a tree graph. Edges in this graph correspond to chosen strategies and vertices correspond to the partial solutions obtained after one iteration. The root of the tree corresponds to the initial PRMM.

1. Estimate depths using an arbitrary strategy $\omega^* \in \Omega^*$ where

$$\begin{aligned} \Omega_{\mathcal{F}} &= \{\, \omega \in \Omega \mid \mathcal{F}(\omega) = \max_{\tau \in \Omega} \mathcal{F}(\tau) \,\} \\ \Omega^* &= \{\, \omega \in \Omega_{\mathcal{F}} \mid \mathcal{S}(\omega) = \max_{\tau \in \Omega_{\mathcal{F}}} \mathcal{S}(\tau) \,\} \end{aligned}$$

2. Fill the missing data.

Repeat steps 1 and 2 until R is complete or no data can be filled in. Then factorize a maximum complete submatrix of R.

Algorithm 4: Scene reconstruction using a set of strategies Ω for estimating the depths.

Practically, however, it is not crucial problem that such obtained strategy is only pseudo-optimal because, as will be seen later, it is possible to realize principle 1 so that, for many scenes, only one iteration is performed. The following proposition holds.

Proposition 1 *The more depths known before the filling, the smaller the expected error.*

Proof of proposition 1 inheres in our extension of Jacob's method (see [60, appendix B]). Usage of principle 1 and proposition 1 in order of their designation proved to be a good criterion. We choose the set of strategies which fill the most points, and from this set, we choose those which scale the most points. From the resulting set, an arbitrary strategy can be used.

The criterion will now be described formally. Let ω denote some strategy for estimating the depths and Ω denote some set of strategies. Let $\mathcal{F}(\omega)$ denote the predicted number of newly filled unknown image points during one iteration when ω is used. The strategy, for which $\mathcal{F}(\omega)$ is maximal, is the best strategy according to principle 1. More such strategies often exist. Let $\mathcal{S}(\omega)$ denote the predicted number of estimated depths when ω is used. According to proposition 1, $\mathcal{S}(\omega)$ is maximal for the best strategy. The complete method for scene reconstruction is summarized in algorithm 4.

The usefulness of the concept of predictor functions $\mathcal{F}, \mathcal{S} : \Omega \longrightarrow 0 \ldots mn$ lies in their ability to be evaluated without neither estimating the depths nor data filling. The knowledge of which image points are known or unknown is the only information for the evaluation of $\mathcal{F}$ and $\mathcal{S}$. It is very simple (and fast) but it cannot detect degenerate configurations of points because, in fact, the multi-view tensors are not computed. If it then, when the tensor is computed, turns out that the configuration is degenerate, the second best strategy is used, etc.

To define $\mathcal{F}$ and $\mathcal{S}$, a few symbols have to be introduced. Let logical variable x_p^i be true if and only if the image point $\mathbf{x}_p^i$ is known. Let i and j be as in step 2 of algorithm 3. Let $\mathcal{I}^{ij}$ be true if and only if the data of image i can be used by the filling method consistently with other images [113]. It is only possible if $i = j$ or if images i and j have enough (at least seven) points in common, which are necessary to compute a fundamental matrix uniquely, thus

$$\mathcal{I}^{ij} \equiv \left|\{p \mid x_p^i \ \wedge \ x_p^j\}\right| \geq 7 \quad \vee \quad i = j \tag{6.1}$$

The uniqueness is demanded for the depths consistency with other images. All available points are used for the fundamental matrix estimation. (i) If there are only seven points, the seven-point algorithm [30] is performed.[8] If it provides three real solutions, the fundamental matrix is not unique. (ii) If there are eight points or more, the eight-point algorithm [30] is performed. In this case, degenerate configurations can be easily detected.[9]

[8] Cheirality constraint [132, 31] may help to filter out some solutions. Further, camera calibration can be exploited using five-, six-, or seven-point algorithms, see section 4.2.

[9] robust epipolar geometry estimation is described in section 4.3

The predictor functions depend on the way how projective depths are computed. Let us first define the predictor functions for the alternative $\omega_{cent,c}$ when the depths are computed using a central image c. Let $\mathcal{P}_p^c$ be true if and only if the p^{th} 3D point can be filled in by the filling method when depths were estimated using strategy $\omega_{cent,c}$. To recover a 3D point uniquely from known basis of the PRMM, at least two of its images are needed. Moreover, it can be proven (see theorem 4 in [60, appendix A]) that at least two known depths in each image are needed for the constraints on $\mathcal{B}$. It means that $\mathcal{P}_p^c$ is true if and only if the p^{th} 3D point is seen in at least two images and the corresponding fundamental matrices, which are needed for estimating at least some two depths in the images, can be computed:

$$\mathcal{P}_p^c \equiv \left|\{i \mid \mathcal{I}^{ic} \ \wedge \ x_p^i\}\right| \geq 2 \tag{6.2}$$

Now, predictor functions $\mathcal{F}$ and $\mathcal{S}$ can be defined as follows

$$\begin{aligned} \mathcal{F}(\omega_{cent,c}) &= \left|\{< i,p > \ \mid \ \mathcal{I}^{ic} \ \wedge \ \mathcal{P}_p^c \ \wedge \ \neg x_p^i\}\right| \\ \mathcal{S}(\omega_{cent,c}) &= \left|\{< i,p > \ \mid \ \mathcal{I}^{ic} \ \wedge \ \mathcal{P}_p^c \ \wedge \ x_p^i \ \wedge \ x_p^c\}\right| \end{aligned}$$

Term $\mathcal{I}^{ic} \ \wedge \ \mathcal{P}_p^c$ says whether point $\mathbf{x}_p^i$ can be reconstructed.

Similarly, the predictor functions for alternative ω_{seq} when the depths are computed for a sequence are defined as

$$\begin{aligned} \mathcal{P}_p &\equiv \left|\{i \mid x_p^i\}\right| \geq 2 \\ \mathcal{F}(\omega_{seq}) &= \left|\{< i,p > \ \mid \ \mathcal{P}_p \ \wedge \ \neg x_p^i\}\right| \\ \mathcal{S}(\omega_{seq}) &= \sum_{p \in 1 \ldots n} \operatorname*{maxarg}_{k \in b(p) \ldots m} \bigwedge_{i \in b(p) \ldots k} x_p^i \end{aligned} \tag{6.3}$$

Equation (6.3) says that the points in the longest continuous subtracks have known depths (see algorithm 3).

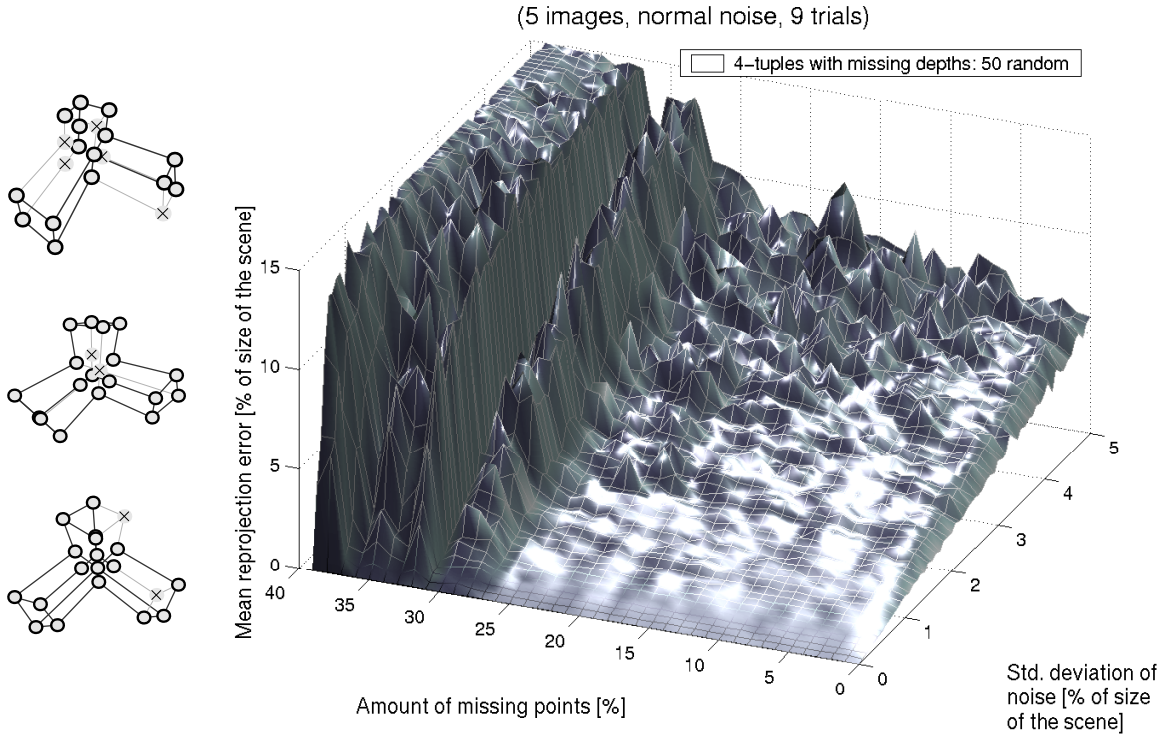
Implementation Details

On account of good numerical conditioning, several normalizations of the data and balancing similar to those in [113] need to be performed. Choosing of quadruplets of columns is implemented so that almost each chosen quadruplet gives the constraint on the basis of the PRMM. This is aimed so that columns are chosen one after another. The columns, which cannot provide the constraint with the already chosen ones, are temporarily removed from the PRMM until the next quadruplet is chosen. By this way, a good efficiency is achieved.

Experiments with Artificial Scenes

For experiments with artificial scenes, a simulated scene with cubes was used. The scene models a real scene, hence it represents a generic situation. Twenty points in space were projected by perspective cameras into several images from different locations and directions. Some image points were made unknown to simulate scene occlusions, see the left-hand side of experiment 1.

Points were taken out from the scene randomly but in a uniform fashion so that, first, the numbers of the missing points in each image differed maximally by one, and secondly, the numbers of images of each point differed maximally by one. Points were only removed as long as the whole scene could still be reconstructed. The necessary condition for a complete reconstruction is that each image contains at least seven points and each point has at least two images (see (6.1) and (6.2)). The more data available, the higher the percentage of missing data permissible. For this specific experiment, i.e. 20 points in 5 images, 65% of missing data is



Experiment 1: Dependency of reprojection error on noise and missing data.

the upper bound allowable to get a complete reconstruction. But because of randomly spread holes in data, the actual level of the maximum amount of the missing data for the complete reconstruction is lower. Experiment 1 shows the dependency of the reprojection error of the reconstruction using algorithm 4 on noise and missing data. Along the left horizontal axis, the amount of the missing data grows while along the right horizontal axis, standard deviation of Gaussian noise of zero mean value added to image points increases. The standard deviation of the added noise as well as the reprojection error is displayed in percentage of the image size.

If no noise is present, the reconstruction is precise. The reprojection error grows linearly with noise with slope approximately equal to one and is almost constant in the direction of the missing data up to the level of missing data above which the reconstruction fails. To conclude, the presented algorithm is accurate and robust with respect to noise as well as missing data.

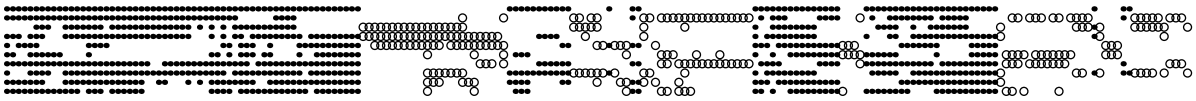
Experiments with Real Scenes

For each experiment, one image, an error table, and the structure of the PRMM are provided. The correspondences across the images have been detected either manually or by the Harris interest operator [28]. Besides the scene name and point detection, the table includes the chosen strategy for estimating the depths, the amount of missing data, the number of images used, image sizes, the number of known points in each image, and reprojection errors for our method in algorithm 4 and projective bundle adjustment[10] initialized by the output of our method. The structure of the PRMM shows the exploitation of image points with known ("•") and unknown ("∘") projective depths. Empty places stand for unknown points. All scenes have been reconstructed in one iteration of algorithm 4.

The "House" scene (see experiment 2) was captured on 10 images in a high resolution. Approximately 100 points were manually detected in each image. Although 47.83% data was

[10] see section 5.4.1

Method	LM = linear method, BA = bundle adj.									
	Scene name	*House*								
	Point detection	manual								
	Depth estimation	$\omega_{cent,1}$								
	Amount of missing data	**47.83%**								
LM	Mean error per image point [pxl]	**3.91**								
LM + BA		**1.44**								
	Image [2952×2003]	1	2	3	4	5	6	7	8	9	10
	Number of corresp.	116	112	97	112	91	79	130	126	101	95
LM	Maximum error [pxl]	11.0	36.6	12.1	9.3	25.8	15.5	13.6	8.9	14.7	13.4
LM + BA		4.3	6.6	4.5	4.4	5.8	8.3	7.5	6.3	10.7	10.1
LM	Mean error	2.3	6.8	3.2	2.3	8.1	5.0	2.5	2.3	3.3	4.8
LM + BA		1.1	1.8	1.5	1.2	1.5	1.6	1.2	1.4	1.5	1.8



size = 10 × 203, " " missing (47.83%), "•" scaled (75.7%), "○" not scaled (24.3%)

Experiment 2: House

Method	LM = linear method, BA = bundle adj.									
	Scene name	*Dinosaur (Oxford)*								
	Point detection	Harris' operator								
	Depth estimation	ω_{seq}								
	Amount of missing data	**90.84%**								
LM	Mean error per image point [pxl]	**1.76**								
LM + BA		**0.64**								
	Image [720×576]	1	5	9	13	17	21	25	29	33	36
	Number of corresp.	257	318	322	516	535	568	602	459	464	381
LM	Maximum error [pxl]	18.4	16.3	29.5	56.4	46.9	73.9	44.1	28.5	19.4	33.9
LM + BA		10.9	12.7	7.8	41.5	25.7	13.1	13.4	17.3	17.9	21.4
LM	Mean error	0.6	0.7	2.3	2.0	3.8	1.7	1.4	1.6	1.3	1.0
LM + BA		0.3	0.5	0.6	1.0	1.0	0.4	0.3	0.5	0.9	0.7

size = 36 × 4983, " " missing (90.84%), "•" scaled (100.0%)

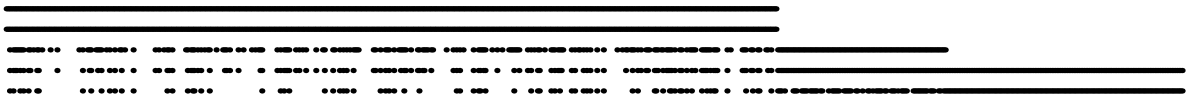
Experiment 3: Dinosaur (Oxford)

missing, the reprojection error, given in pixels, is low considering the image sizes. It can be seen that our algorithm could have exploited all known data including 24.3 % unscaled points.

The "Dinosaur" scene (see experiment 3) was captured on 36 images. Points were detected automatically by the Harris operator. Although the amount of missing data is high (90.84%), the mean error per image point was lower because of more precise point detection and since 100% of points were scaled. It turned out later in [65] (section 6.3.1) after doing the metric upgrade that the projective reconstruction obtained using this method is wrong (see also footnote 11).

The data in experiment 4 contained outliers, that were removed one after another in the following manner. The scene was first reconstructed with all the data including outliers. Then, the column of the PRMM, which contained the point with the highest reprojection error, was discarded. Afterwards, the scene was reconstructed again, another column discarded etc. These

Method	LM = linear method, BA = bundle adj.					
	Scene name	*Temple (Leuven)*				
	Point detection	Harris' operator				
	Depth estimation	ω_{seq}				
	Amount of missing data				**46.32%**	
LM	Mean error per image point [pxl]				**0.49**	
LM + BA					**0.23**	
	Image [867×591]	1	2	3	4	5
	Number of corresp.	456	456	297	374	285
LM	Maximum error [pxl]	3.0	2.3	2.8	2.5	3.0
LM + BA		2.5	1.5	2.4	1.8	2.5
LM	Mean error	0.4	0.5	0.6	0.5	0.5
LM + BA		0.3	0.2	0.2	0.2	0.2



size = 5 × 696, " " missing (46.32%), "•" scaled (100.0%)

Experiment 4: Temple (Leuven)

two steps were repeated till the highest reprojection error was significant. For the "Temple" scene in experiment 4, the threshold was set to 4 pixels which lead to discarding 23 out of 719 columns.

To conclude, the presented algorithm is enough accurate on real scenes to provide a good initial solution for bundle adjustment. [11]

Summary and Conclusions

A linear method for scene reconstruction has been presented and tested on artificial and real scenes. The method extends and suitably combines previous methods so that the reconstruction in an entirely general situation, i.e. many images with perspective camera and occlusions, is possible.

A way of exploiting points with unknown depth was developed. Correctness of this way was proved as well as its abilities and limitations [60]. Its theoretical asset is the ability to reconstruct linearly some very small scene configurations, which can be reconstructed by other methods only nonlinearly (see theorem 3 in [60]), cannot be reconstructed at all (see theorem 2 in [60]), or cannot exploit all known data (see theorem 1 in [60]). Moreover, it gives good results in practical situations as presented here.

The presented method was intended to deal with several problems in 3D reconstruction. These were the perspective projection, many images, and occlusion. However, one problem was not taken into account explicitly and that is the problem of outliers in correspondences. Although the method was not intended to deal with outliers, it was observed that it can deal with them if they are few compared to the number of inliers (see experiment 4). To deal well with a bigger amount of outliers, extension [39] of factorization handling outliers can be added. [12]

[11] After publishing [62] (section 6.1), it turned out that the metric upgrade of the Dinosaur sequence was not possible because the projective reconstruction obtained using this method is wrong. High maximum residuals in experiment 3 indicate that something is wrong. Reformulation of the Jacobs' method [43] in the original subspaces was proposed in [65] (section 6.3).

[12] see footnote 11

6.2 Mismatch Detection by Trifocal Tensor Voting

We tried to detect mismatches which passed the EG test before placing the EG inliers into the MM. In [61], we used a naive RANSAC on trifocal tensors, which will be explained below.

The method assumes that the amount of inliers is significantly larger than the amount of outliers. This is true[13] thanks to matcher [69] that discards all matches that do not satisfy consistent epipolar geometries between image pairs. The main idea is that minimal configurations of points in triplets of images are sufficient to validate inliers reliably. The RANSAC paradigm is used. Trifocal tensors are computed from randomly selected minimal 6-tuples of points in triplets of images using method [102]. After the tensor estimation, the number of points consistent with the tensor is counted. If there are sufficiently enough consistent points (controlled by a threshold), those not used to estimate the trifocal tensor receive one positive vote. The voting is repeated until points in the measurement matrix are sufficiently sampled. The points that obtain zero or a very small number of votes are rejected as outliers. Inliers are used by the method described in [62] (section 6.1) to obtain a projective reconstruction. The set of inliers can be further enlarged by an iterative process.

Method [61] can inspect correspondences among at least three images because the trifocal tensors are used. However, there may be some correct two-view correspondences as well, which cannot appear at the voting stage. Each image point marked by method [61] as outlier is finally verified by sampling pair-wise correspondences in the corresponding column of MM. Each such sample is checked by reconstructing the 3D point using the known cameras. If the reprojection errors in both images are small, the correspondence (and the image point) is validated.

Method [61] is not presented here in full detail as it turned out to be almost unusable in practice (this refers to practically all scenes presented in the following sections) and no further work built on it.

[13] not on difficult scenes such as those in section 6.8.

6.3 Projective Gluing

This section[14] describes a technique for building consistent 3D reconstructions from many views based on fitting a low rank matrix to a matrix with missing data. Rank-four submatrices of minimal, or slightly larger, size are sampled and spans of their columns are combined to constrain a basis of the fitted matrix. The error minimized is expressed in terms of the original subspaces which leads to better resistance to noise compared to previous methods. More than 90% of the missing data can be handled while finding an acceptable solution efficiently. Applications to 3D reconstruction using both affine and perspective camera models are shown. For the perspective model, a linear method based on logarithms of positive depths from cheirality is presented to make the depths consistent with an overdetermined set of epipolar geometries. Results are shown for scenes and sequences of various types. Many images in open and closed sequences in narrow and wide baseline setups are reconstructed with reprojection errors around one pixel. It is shown that reconstructed cameras can be used to obtain dense reconstructions from epipolarly aligned images.

Several attempts to provide reconstruction from many images in a one-step algorithm have been made (see section 3.1). However, (before 2005) none of these methods succeeded to process more than a few tens of images when the amount of missing elements reaches 90% of the measurement matrix and cameras have large field of view or are in a wide baseline setup. In this section we present a technique that builds a consistent reconstruction (1) for scenes of various types: open and closed sequences in both narrow and wide baseline setups, and (2) for various camera models: affine and perspective, which can model also omni-directional cameras [73].

Our algorithm has the following advantages: (i) it provides an overall scene structure and motion in a single step without requirements such as linear ordering of images in a sequence (ii) the solution is obtained as a global optimum of a reasonable cost function defined on an approximation to the original SFM (structure-from-motion) problem. The obtained projective reconstruction can be easily upgraded to the metric one, see figure 6.3. The result can be used for dense reconstruction, see figure 6.10.

This method cannot be classified as a factorization method, although factorization on small complete matrices appears inside. Similarly to [27], it produces a direct solution on camera matrices. In contrast to [27], the minimized error is expressed in terms of image data and not elements of fundamental matrices as in [27].

Our method does not try to fill any elements and even does not hallucinate them as, e.g., in [10][15]. The missing data are not modeled in the algorithm at all. Only known data are exploited while minimizing their distance to the fitted matrix. The method bootstraps from rank-four submatrices of minimal or slightly larger size. Linear spaces generated by their rows or columns are combined to constrain the basis of the whole measurement matrix. This idea has already appeared in [43]. However, the way it is realized here is novel.

The most crucial difference from [43] is that the solved problem is formulated in terms of the original subspaces, and not the complementary ones as in [43]. Therefore, error due to noise is corrected where it was physically caused, i.e. in the spaces generated by image measurements and not in their complements. Our formulation is equivalent to [43] in case there is no noise in the data. However, as a reasonable error is minimized, it has much better behaviour when noise is present which enables handling lots of missing data. Moreover, it leads to precise and fast algorithms as only a small system of equations with a sparse design matrix has to be solved. In application to 3D reconstruction, large data compression is reached by taking only the four

[14] This section is an extended version of [65]. Andrew Zisserman from the University of Oxford kindly provided the Dinosaur data, Tomáš Werner from the Czech Technical University (CTU) provided the routine for the bundle adjustment and Jana Kostková from CTU kindly made the dense reconstructions.

[15] Method [10] was not usable on the data presented in this section at all.

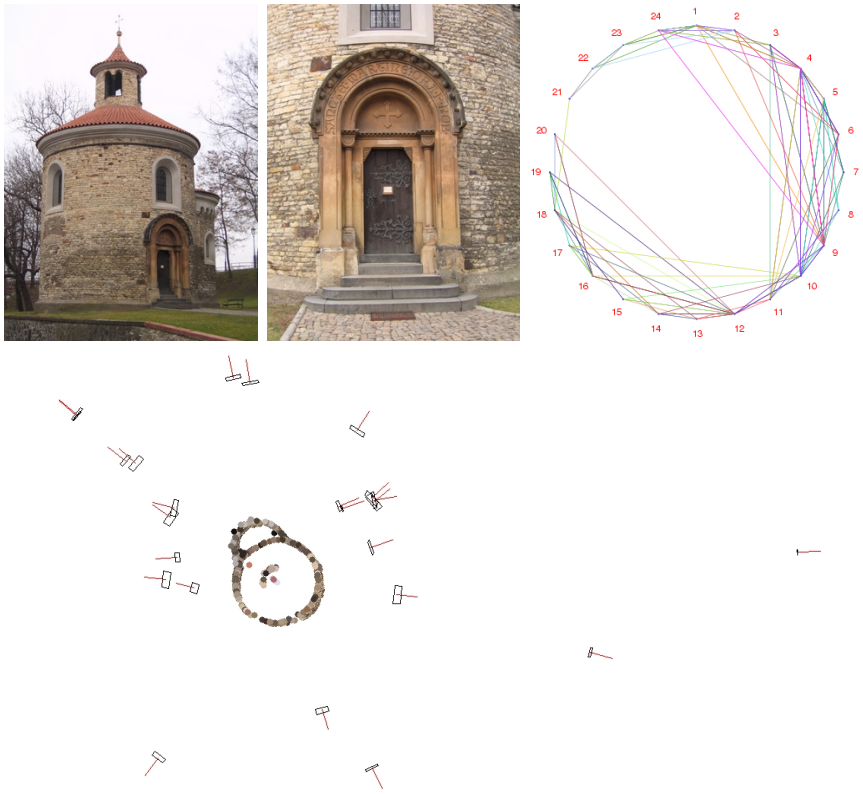


Figure 6.3: Reconstruction from a wide baseline scenario after the metric BA: the St. Martin rotunda on 24 images, 89% data missing, top view. Some cameras are positioned very close to each other while some are distant making the SFM problem difficult.

singular vectors best explaining the submatrix of (possibly) large amount of points seen in an image pair or triplet.

This section describes: (i) a technique for fitting a low-rank matrix to a matrix with missing data is introduced (section 6.3.1). (ii) Two ways of its application to the structure-from-motion problem are given for both affine and perspective camera models (sections 6.3.4 and 6.3.5), (iii) a method for estimating projective depths consistent with an overdetermined system of epipolar geometries is introduced (section 6.3.6). Section 6.3.7 studies possible ways of fixing some depths. Experiments are reported in section 6.3.8. Metric upgrade is described in section 6.3.9.

6.3.1 Fitting Matrices with Missing Data

Let $\mathbf{y} \in \mathbb{R}^{3m \times n}$ be some matrix with missing elements. The method will be explained on rank-four matrices, but it can be used for any rank. Matrix $\mathbf{y}$ can represent, e.g., the measurement matrix (MM, *p*7) rescaled by the projective depths, $[\lambda_p^i \mathbf{x}_p^i]_{p=1\ldots n}^{i=1\ldots m}$. The task is to find the best rank-four fit $\mathtt{PX}$ to known elements of $\mathbf{y}$ in the least squares where $\mathtt{P} \in \mathbb{R}^{3m \times 4}$, $\mathtt{X} \in \mathbb{R}^{4 \times n}$. A description of a good suboptimal solution follows.

Rank-four submatrices of $\mathbf{y}$ will be used to constrain the column basis $\mathtt{P}$ of $\mathbf{y}$. Let their number be denoted by T. Let $\mathbf{i}_t$ and $\mathbf{p}_t$ denote sets of row triplet (camera) and column indices of the t^{th} submatrix within $\mathbf{y}$, respectively, $3|\mathbf{i}_t| \geq 4$, $|\mathbf{p}_t| = 4$. Notation $\mathtt{A}_{\mathbf{p}}^{\mathbf{i}}$ will denote the submatrix of $\mathtt{A}$ composed of elements in rows $\mathbf{i}$ and columns $\mathbf{p}$. Omitting superscript or subscript means taking all rows or columns, respectively. Let only submatrices with (i) all its elements known and (ii)

linearly independent columns be chosen. Let the t^{th} submatrix be denoted by $\tilde{\mathtt{P}}_t$, $\tilde{\mathtt{P}}_t = \mathbf{y}^{\mathbf{i}_t}_{\mathbf{p}_t}$.[16] Then,

$$\begin{aligned} \tilde{\mathtt{P}}_1 &= \mathtt{P}^{\mathbf{i}_1}\mathtt{X}_{\mathbf{p}_1} \\ &\vdots \\ \tilde{\mathtt{P}}_T &= \mathtt{P}^{\mathbf{i}_T}\mathtt{X}_{\mathbf{p}_T}. \end{aligned} \tag{6.4}$$

From linear independency of columns of $\tilde{\mathtt{P}}_t$, $|\mathbf{p}_t| = 4$, and (6.4),

$$\operatorname{rank}\tilde{\mathtt{P}}_t = \operatorname{rank}\mathtt{P}^{\mathbf{i}_t} = \operatorname{rank}\mathtt{X}_{\mathbf{p}_t} = 4. \tag{6.5}$$

Jacobs [43] used the fact that $\operatorname{span}\tilde{\mathtt{P}}_t = \operatorname{span}\mathtt{P}^{\mathbf{i}_t}$ to constrain $\mathtt{P}$ by $\operatorname{span}\mathtt{P} \subseteq \operatorname{span}\mathbf{y}_{\mathbf{p}_t}$ where $\operatorname{span}\mathbf{y}_{\mathbf{p}_t}$ can be interpreted as the linear hull of all possible fillings of $\mathbf{y}_{\mathbf{p}_t}$ (see the interpretation in figure 6.1, *p45*). He formulated the constraint using complementary subspaces $\mathtt{N}_t = (\operatorname{span}\mathbf{y}_{\mathbf{p}_t})^\perp$ as $\mathtt{P} \subseteq \mathtt{N}^\perp$ where $\mathtt{N}$ is the union of the complementary subspaces, $\mathtt{N} = \bigcup_{t=1,\dots,T}\mathtt{N}_t$.

However, this formulation does not treat noise well. Small changes in $\mathtt{N}_t$ (caused by noise in $\mathbf{y}_{\mathbf{p}_t}$) are accumulated in their union $\mathtt{N}$ and may result into a large change in $\mathtt{N}^\perp$. The reason is that the noise is physically caused in the original subspaces (on image data) where it should also be corrected, as our method does, which will be shown below. In fact, [43] corrects the error in the complementary subspaces ($\mathtt{N}_t$, $\mathtt{N}$) which have an unclear connection to the original noise.[17] We observed that [43] breaks down when noise is present and the number of images, over which the partial reconstructions are glued, reaches some limit. E.g., [43] could reconstruct only a subsequence of at most 22 images of the Dinosaur sequence shown in figure 6.4. We have not observed any such limit at our method even when hundreds of images were used.

Our approach exploits the fact that $\operatorname{rank}\mathtt{X}_{\mathbf{p}_t} = 4$ thanks to which the inverse to $\mathtt{X}_{\mathbf{p}_t}$ exists, $\mathtt{H}_t = \mathtt{X}^{-1}_{\mathbf{p}_t}$. The t^{th} equation in (6.4) can be now multiplied by $\mathtt{H}_t$ from the right:

$$\begin{aligned} \tilde{\mathtt{P}}_1\mathtt{H}_1 &= \mathtt{P}^{\mathbf{i}_1} \\ &\vdots \\ \tilde{\mathtt{P}}_T\mathtt{H}_T &= \mathtt{P}^{\mathbf{i}_T}. \end{aligned} \tag{6.6}$$

Although equations (6.4) are bilinear in unknowns $\mathtt{P}$ and $\mathtt{X}$, equations (6.6) are linear in all unknowns, $\mathtt{P}$ and $\mathtt{H}_t$, and thus, given sufficiently many equations, solvable uniquely up to an overall projective transformation. Due to the bilinearity of the original problem (6.4), only its approximation is found by solving the transformed problem (6.6). However, it is a good approximation, as will be demonstrated.

Solving System (6.6)

Denoting $\mathtt{P}^{\mathbf{i}_t} = [\mathbf{q}^{\mathbf{i}_t}_1\mathbf{q}^{\mathbf{i}_t}_2\mathbf{q}^{\mathbf{i}_t}_3\mathbf{q}^{\mathbf{i}_t}_4]$ and $\mathtt{H}_t = [\mathbf{h}_{t,1}\mathbf{h}_{t,2}\mathbf{h}_{t,3}\mathbf{h}_{t,4}]$, the t^{th} equation in (6.6) can be rewritten as

$$\begin{aligned} \tilde{\mathtt{P}}_t\mathbf{h}_{t,1} - \mathbf{q}^{\mathbf{i}_t}_1 &= 0 \\ \tilde{\mathtt{P}}_t\mathbf{h}_{t,2} - \mathbf{q}^{\mathbf{i}_t}_2 &= 0 \\ \tilde{\mathtt{P}}_t\mathbf{h}_{t,3} - \mathbf{q}^{\mathbf{i}_t}_3 &= 0 \\ \tilde{\mathtt{P}}_t\mathbf{h}_{t,4} - \mathbf{q}^{\mathbf{i}_t}_4 &= 0. \end{aligned} \tag{6.7}$$

[16] $\tilde{\mathtt{P}}_t$ can be viewed as cameras in a projective reconstruction of $\mathbf{y}^{\mathbf{i}_t}_{\mathbf{p}_t}$ with points $\mathtt{I}_{4\times4}$, $\mathbf{y}^{\mathbf{i}_t}_{\mathbf{p}_t} = \tilde{\mathtt{P}}_t\mathtt{I}_{4\times4}$, where $\mathtt{I}$ denotes the identity matrix.

[17] Subspace $\mathtt{N}_t$ is represented in [43] using an orthonormal basis of the complementary subspace to $\operatorname{span}\mathbf{y}_{\mathbf{p}_t}$.

Denoting $\mathbf{z}_c = (\mathbf{h}_{1,c}, \ldots, \mathbf{h}_{T,c}, \mathbf{q}_c^{\mathbf{i}_1}, \ldots, \mathbf{q}_c^{\mathbf{i}_T})^\top$, the whole system (6.6) can be rewritten as

$$\underbrace{\begin{bmatrix} \mathtt{A} & \mathtt{0} & \mathtt{0} & \mathtt{0} \\ \mathtt{0} & \mathtt{A} & \mathtt{0} & \mathtt{0} \\ \mathtt{0} & \mathtt{0} & \mathtt{A} & \mathtt{0} \\ \mathtt{0} & \mathtt{0} & \mathtt{0} & \mathtt{A} \end{bmatrix}}_{\mathtt{B}_{4f \times 4g}} \begin{pmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \mathbf{z}_3 \\ \mathbf{z}_4 \end{pmatrix} \quad = \quad \mathtt{0}_{4f \times 1} \tag{6.8}$$

where $\mathtt{A}$ and all $\mathtt{0}$ matrices are of size $f \times g$, $f = \sum_{t=1}^{T} 3|\mathbf{i}_t|$, $g = 4|T| + f$. Matrix $\mathtt{A}$ is composed of T sub-blocks in the form $\mathtt{A}^{\mathcal{I}(\mathbf{i}_t)}_{[4t-3:4t,4|T|+\mathcal{I}(\mathbf{i}_t)]} = \mathtt{C}_t = [\tilde{\mathtt{P}}_t \,|-\mathtt{I}_{3|\mathbf{i}_t| \times 3|\mathbf{i}_t|}]$ corresponding to one equation in (6.7) where $\mathcal{I}(\mathbf{i})$ returns indices of rows in the MM corresponding to images $\mathbf{i}$. Matrix $\mathtt{C}_t$ is of size $3|\mathbf{i}_t| \times (4 + 3|\mathbf{i}_t|)$, $\operatorname{rank} \mathtt{C}_t = 3|\mathbf{i}_t|$, $\dim \operatorname{null} \mathtt{C}_t = 4$. If the partial reconstructions have sufficient overlaps in cameras for ensuring that all cameras $\mathtt{P}^i$ have consistent projective frames[18], $\dim \operatorname{null} \mathtt{A} = 4$ in case there is no noise in the data. Consequently, $\dim \operatorname{null} \mathtt{B} = 16$, see (6.8). The number sixteen corresponds to the freedom for the sixteen parameters of the overall projective transformation. However, not all solutions from this sixteen-dimensional space are acceptable. It is required that the solution satisfies $\operatorname{rank} \mathtt{H}_t = \operatorname{rank} \mathtt{P}^{\mathbf{i}_t} = 4$, see (6.5), which is equivalent conjunction:

$$\left.\begin{array}{r} \mathbf{h}_{t,a} \text{ and } \mathbf{h}_{t,b} \text{ are linearly independent} \\ \mathbf{q}_a^{\mathbf{i}_t} \text{ and } \mathbf{q}_b^{\mathbf{i}_t} \text{ are linearly independent} \end{array}\right\} \text{ for } a \neq b. \tag{6.9}$$

It might be possible to solve the large system (6.8), e.g., by MATLAB's EIGS on $\mathtt{B}^\top\mathtt{B}$, and to choose (we do not know how) some appropriate vector from its sixteen-dimensional solution space. Nevertheless, a more simple and efficient way is to find the best four linearly independent solutions to system

$$\mathtt{A}\mathbf{z} \quad = \quad \mathtt{0}_{f \times 1} \tag{6.10}$$

in the least squares. These solutions satisfy properties (6.9):

Proposition 2 *Let $\mathbf{z}_1$, $\mathbf{z}_2$, $\mathbf{z}_3$, $\mathbf{z}_4$ be four linearly independent solutions to system (6.10). Then, (6.9) holds.*

Proof. The idea of the proof is shown for $T = 1$. The solution for $T > 1$ is analogical. Let the assumption hold. For contradiction, let, e.g., $\mathbf{h}_{t,2} = \alpha \mathbf{h}_{t,1}$ for some $\alpha \in \mathbb{R}$. Then, using (6.7), columns of

$$\begin{bmatrix} \mathbf{h}_{t,1} & \mathbf{h}_{t,2} \\ \mathbf{q}_1^{\mathbf{i}_t} & \mathbf{q}_2^{\mathbf{i}_t} \end{bmatrix} = \begin{bmatrix} \mathbf{h}_{t,1} & \mathbf{h}_{t,2} \\ \tilde{\mathtt{P}}_t \mathbf{h}_{t,1} & \tilde{\mathtt{P}}_t \mathbf{h}_{t,2} \end{bmatrix} = \begin{bmatrix} \mathbf{h}_{t,1} & \alpha \mathbf{h}_{t,1} \\ \tilde{\mathtt{P}}_t \mathbf{h}_{t,1} & \alpha \tilde{\mathtt{P}}_t \mathbf{h}_{t,1} \end{bmatrix}$$

are linearly dependent. Contradiction.

In case of noisy data, $\tilde{\mathtt{P}}_t$ in the equation above can be replaced by another rank-four matrix, $\dot{\mathtt{P}}_t$, close to $\tilde{\mathtt{P}}_t$ such that equations (6.7) hold for $\dot{\mathtt{P}}_t$ exactly. □

It turned out in our experiments that transforming $\tilde{\mathtt{P}}_t$ into an orthonormal basis by $\tilde{\mathtt{P}}_t \mapsto \tilde{\mathtt{P}}_t \mathtt{G}_t$ is a good choice, as in [43]. Note that $\mathtt{G}_t$ is absorbed by the estimated $\mathtt{H}_t$ matrix achieving well conditioning of the system.

[18] Let the projective frame of cameras $\mathbf{i}_t$ be chosen as a reference frame for some t. Camera a has a consistent frame if there is (i) one image triplet $\mathbf{i}_t = \{a, b, c\}$ or (ii) two image pairs $\mathbf{i}_t = \{a, b\}$ and $\mathbf{i}_{t'} = \{a, c\}$ such that b and c have a consistent frame and the three camera centers are not colinear (see discussion in section 6.7.6, p96).

6.3.2 What Is Being Minimized

The best approximate solution to the original problem (6.4) in the least square sense minimizes error

$$e_{orig} = \min_{\operatorname{rank} \mathtt{P}^{\mathbf{i}t}=\operatorname{rank} \mathtt{X}_{\mathbf{p}_t}=4} \sum_t \left\| \tilde{\mathtt{P}}_t - \mathtt{P}^{\mathbf{i}_t} \mathtt{X}_{\mathbf{p}_t} \right\|^2$$

where $\|\cdot\|$ denotes the Frobenius norm. In (6.6), error

$$er_{min} = \min_{\operatorname{rank} \mathtt{P}^{\mathbf{i}t}=\operatorname{rank} \mathtt{H}_t=4} \sum_t \left\| \tilde{\mathtt{P}}_t \mathtt{H}_t - \mathtt{P}^{\mathbf{i}_t} \right\|^2 \tag{6.11}$$

is minimized. Remember that t goes over all sampled rank-four submatrices thus in a typical situation the same elements of $\mathtt{P}$ appear many times in the previous formula. Therefore, in presence of noise it is impossible to reach zero value for er_{min}. Although er_{min} differs from e_{orig}, it is still reasonable to minimize such error, as will be shown in experiments.

Remind that factorization minimizes exactly the reprojection error when the affine camera model is used. Using the perspective camera model, if all the depths are close to equal, then it minimizes a good approximation to the reprojection error [31, *p*446]. Factorization searches for such a four-dimensional subspace that best approximates each data column. In our framework, an extensive sampling of matrices $\mathbf{y}^{\mathbf{i}_t}_{\mathbf{p}_t}$ would be necessary for reaching a similar effect at least for that all the data is used.

Such sampling would be computationally expensive and it is not clear to us how successful could be such an attempt in, e.g., equiponderant exploitation of all the data. Nevertheless, there is a way of simplifying the sampling. Its idea comes out from that whatever data are contained in matrices $\tilde{\mathtt{P}}_t$, their columns are always transformed (close) to $\mathtt{P}^{\mathbf{i}_t}$, see (6.6). So if two matrices $\tilde{\mathtt{P}}_t$ and $\tilde{\mathtt{P}}_s$, $t \neq s$, share the same row (or column) indices, $\mathbf{i}_t = \mathbf{i}_s$, it makes sense to use the least squares approximation to the two subspaces instead. The effect is not only reduction of the number of unknowns $\mathtt{H}_t$ but foremost suppression of noise. The least squares approximation can be obtained by factorization using SVD. By this, the powerful feature of factorization of the optimal propagation of error is adopted. It is good to use factorization globally. In our framework, factorization is used only locally due to the missing data but it does not matter much because a global propagation of error is done in (6.6). This propagation is done very finely as the solution is searched for in a high dimensional space thanks to many auxiliary variables $\mathtt{H}_t$, see (6.11). Thus, our model is very rich compared to [43] but it does not overfit. Consistent cameras $\mathtt{P}$ and homographies $\mathtt{H}_t$ are searched for so that projections of the four points $\mathtt{I}_{4\times4}$ best fit all partial reconstructions, $\tilde{\mathtt{P}}_t\mathtt{H}_t = \mathtt{P}^{\mathbf{i}_t}\mathtt{I}_{4\times4}$.

6.3.3 Aligning Partial Reconstructions

It would be best to estimate the subspaces from as large submatrices as possible. However, finding the largest complete submatrix in a matrix with missing elements is known to be NP-hard [43]. Moreover, in vision applications, image measurements often originate from image pairs or image triplets. Therefore, we use all known measurements in an image pair and triplet, similarly to [43]. Thus, for given image indices $\mathbf{i}$, submatrices $\mathbf{y}^{\mathbf{i}}_{\mathbf{p}}$ are as wide as possible.

Outlier rejection from such matrices can be easily done using, e.g., RANSAC or iterative factorization with rejecting points with reprojection errors above some threshold after each iteration (both ways lead to similar results in our experiments with 1 pxl threshold). The column basis of span $\mathbf{y}^{\mathbf{i}_t}_{\mathbf{p}_t}$, denote it by $\hat{\mathtt{P}}_t$, is estimated as $\hat{\mathtt{P}}_t = \mathtt{U}_{1,2,3,4}$ where $\mathbf{y}^{\mathbf{i}_t}_{\mathbf{p}_t} = \mathtt{U}\operatorname{diag}(\sigma_1, \ldots, \sigma_z)\mathtt{V}^\top$ is the SVD factorization. The row basis of span $\mathbf{y}^{\mathbf{i}_t}_{\mathbf{p}_t}$, $\hat{\mathtt{X}}_t$, is estimated as $\hat{\mathtt{X}}_t = {\mathtt{V}_{1,2,3,4}}^\top$.

Partial reconstructions are aligned via cameras using:

$$\begin{aligned} \omega_1 \hat{\mathtt{P}}_1 \mathtt{H}_1 &= \omega_1 \mathtt{P}^{\mathbf{i}_1} \\ &\vdots \\ \omega_T \hat{\mathtt{P}}_T \mathtt{H}_T &= \omega_T \mathtt{P}^{\mathbf{i}_T}. \end{aligned} \tag{6.12}$$

Here, ω_t denotes the weight of the t^{th} partial reconstruction taking into consideration belief of the estimate of the reconstruction expressed in terms of the number of correspondences consistent with it:

$$\omega_t = \sqrt{\frac{n_t}{\bar{n}}}$$

where $n_t = |\mathbf{p}_t|$ and $\bar{n}$ is the average number of correspondences. Normalization by $\bar{n}$ gets all weights close to one and is done due to conditioning. System (6.12) is solved in the least squares, thus the square root from ω_t disappears in the minimized error, see (6.11), which thanks to this well approximates the reprojection error measured on the whole MM.

Aligning reconstructions via points, the transposed problem, is solved similarly using the row bases:

$$\begin{aligned} \omega_1 \hat{\mathtt{X}}_1^\top \tilde{\mathtt{H}}_1 &= \omega_1 \mathtt{X}_{\mathbf{p}_1}^\top \\ &\vdots \\ \omega_T \hat{\mathtt{X}}_T^\top \tilde{\mathtt{H}}_T &= \omega_T \mathtt{X}_{\mathbf{p}_T}^\top. \end{aligned} \tag{6.13}$$

Here, n_t used to compute ω_t denotes the number of cameras, $n_t = |\mathbf{i}_t|$.

Indeed, system (6.12) can be interpreted as aligning or gluing partial reconstructions, each represented in a different projective coordinate frame by at least two cameras $\hat{\mathtt{P}}_t$, $|\mathbf{i}_t| \geq 2$. Homography $\mathtt{H}_t$ maps the coordinate system of the t^{th} partial reconstruction to the global coordinate system of the reconstruction of all data. Similarly, in system (6.13) each partial reconstruction is represented by at least four points $\hat{\mathtt{X}}_t$, $|\mathbf{p}_t| \geq 4$. Systems (6.12) and (6.13) are special cases of (6.6).

It would be desirable to do allignment using both cameras and points simultaneously[19] as one could expect achieving a higher accuracy thanks to exploitation of more constraints. Unfortunatelly, combining of equations (6.12) and (6.13) is non-trivial due to the non-linearity of the relation between the corresponding transformations: $\mathtt{H}_t = (\tilde{\mathtt{H}}_t)^{-\top}$. This relation becomes linear only in case when $\mathtt{H}_t$ is a orthonormal as then the matrix inverse becomes transposition and consequently $\mathtt{H}_t = \tilde{\mathtt{H}}_t$. However, treating $\mathtt{H}_t$ as orthonormal seems to be wrong (we have not tried that in experiments). Nevertheless, this technique seems to be usable provided that camera internals are known and after applying the following modification of the data representing the partial reconstructions. The data in the i^{th} image, $\mathbf{y}^i$, are transformed to $\mathtt{K}^{i\,-1}\mathbf{y}^i$ where $\mathtt{K}^i \in \mathbb{R}^{3\times 3}$ are camera internals of the i^{th} camera. Then, it should be possible (we have not tried that) to modify the decompositions $\mathbf{y}_{\mathbf{p}_t}^{\mathbf{i}_t} = \hat{\mathtt{P}}_t \hat{\mathtt{X}}_t$ so that the first three columns of $\hat{\mathtt{P}}_t$ are two rotation matrices stacked on top of each other. Then, trasformations $\mathtt{H}_t$ and $\tilde{\mathtt{H}}_t$ are simplified to

$$\mathtt{H}_t = \begin{bmatrix} \mathtt{R}_t & \mathbf{t}_t \\ \mathtt{0}_{1\times 3} & s_t \end{bmatrix} \quad \text{and} \quad \tilde{\mathtt{H}}_t^\top = \begin{bmatrix} \mathtt{R}_t{}^\top & -\mathtt{R}_t{}^\top \mathbf{t}_t \frac{1}{s_t} \\ \mathtt{0}_{1\times 3} & \frac{1}{s_t} \end{bmatrix} \tag{6.14}$$

where $\mathtt{R}_t$ is a rotation, $\mathbf{t}_t$ is a translation and s_t is a scale. One can either (i) use all equations (6.12) and (6.13) with substitutions (6.14) where the non-linear terms in the last column of $\tilde{\mathtt{H}}_t^\top$ are replaced by some new variables. Or, (ii) one can first solve for rotations using only a subsystem of equations (6.12) and (6.13) involving only the upper left 3×3 block of $\mathtt{H}_t$ and

[19] Marc Pollefeys, one of the thesis reviewers, asked for a comment on this.

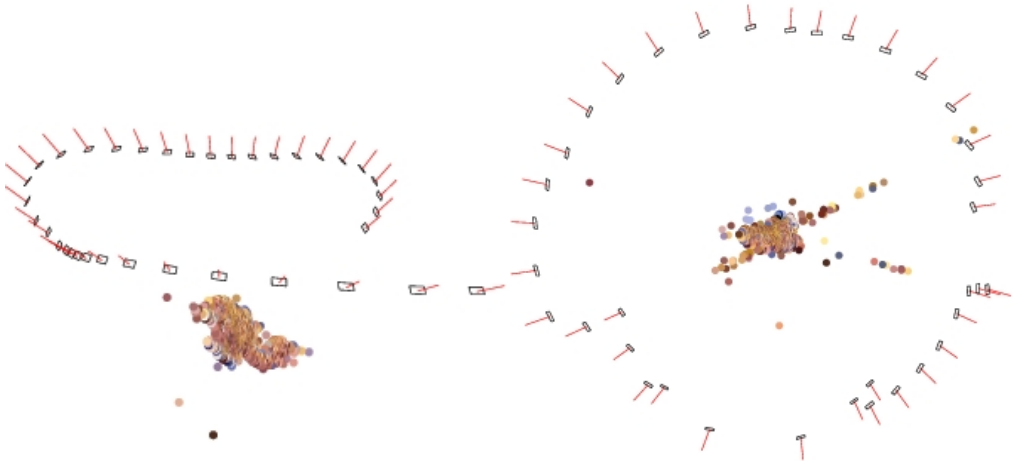


Figure 6.4: Initial reconstruction of the Dinosaur sequence of 35 images using affine camera model and gluing via points: Open sequence with mean reprojection error 2.57 pxl. 72 image triplet constraints were used (left) Closed sequence with mean reprojection error 2.65 pxl. 74 image triplet constraints were used (right).

$\tilde{\mathtt{H}}_t$ in (6.14) and then solving for translations, using, e.g., the Kahl's method [44] as it was done in [66] (section 6.7.2).

To demonstrate the quality of approximation to (6.4) by (6.12), comparison with factorization by SVD on a complete MM was done. Our method gave only 0.7% worse mean reprojection error than SVD of the complete MM obtained by multiplying cameras $\mathtt{P}$ and points $\mathtt{X}$ from the reconstruction of the Dinosaur sequence with added 1 pxl noise.[20] Note that in contrast to factorization our approach can handle the missing data.

It turned out in our experiments that adding also constraints from image four-, five-, ...-tuples did not improve results much. The reasons are the following: (i) the less images, the more matches are in them and thus the better estimation of $\hat{\mathtt{P}}_t$ and $\hat{\mathtt{X}}_t$ and (ii) the more images, the more it is likely that an outlier appears in a track (column of $\mathbf{y}^{\mathbf{i}}_{\mathbf{p}}$) regardless of the type of the used outlier rejection scheme.

6.3.4 Affine Camera Model

In the affine model, camera centers are considered to be infinitely distant from the scene structure, hence (i) all depths are equal and can be set to one and (ii) the last row of all camera matrices is $[\ 0\,0\,0\,1\ ]$. Therefore, image projection equation (2.1), *p*7 can be rewritten as

$$1\begin{pmatrix} \bar{\mathbf{x}}^i_p \\ 1 \end{pmatrix} = \left[\begin{array}{c|c} \bar{\mathtt{P}}^i & \mathbf{t}^i \\ 0\,0\,0 & 1 \end{array}\right] \begin{pmatrix} \bar{\mathbf{X}}_p \\ 1 \end{pmatrix}$$

and simplified to

$$\bar{\mathbf{x}}^i_p = \left[\ \bar{\mathtt{P}}^i\,\mathbf{t}^i\ \right] \begin{pmatrix} \bar{\mathbf{X}}_p \\ 1 \end{pmatrix}. \tag{6.15}$$

Let $\bar{\mathbf{x}}^{\mathbf{i}_t}_{\mathbf{p}_t} = \mathtt{U}\,\mathrm{diag}(\sigma_1, \ldots, \sigma_z)\mathtt{V}^\top$ be the SVD factorization.

[20] The perspective camera model was used. It was used in this section with Hartley's normalization of the image measurements. Here, the projective depths were obtained from the reconstruction. Partial reconstructions from image triplets 1-2-3, 2-3-4, ..., $(m-2)-(m-1)-m$ were used.

(i) *Gluing via cameras.* Equations (6.12) are of the following form due to the special structure of affine homographies $\mathtt{H}_t$:

$$[\hat{\bar{\mathtt{P}}}_t \hat{\mathbf{t}}_t] \begin{bmatrix} \mathtt{A}_t & \mathbf{b}_t \\ \mathtt{0}_{1\times 3} & 1 \end{bmatrix} = [\bar{\mathtt{P}}^{\mathbf{i}_t} \mathbf{t}^{\mathbf{i}_t}] \quad \mathrm{t} = 1,\ldots,\mathrm{T}.$$

From this, $\hat{\bar{\mathtt{P}}}_t \mathtt{A}_t = \bar{\mathtt{P}}^{\mathbf{i}_t}$, which allows to estimate $\bar{\mathtt{P}}$ up to translations as a rank-three fit similarly to (6.12), provided $\hat{\bar{\mathtt{P}}}_t$ have been estimated as $\hat{\bar{\mathtt{P}}}_t = \mathtt{U}_{1,2,3}$. Then, translations of all cameras $\mathbf{t}^i$ and all points $\bar{\mathbf{X}}_p$ can be estimated from the non-homogenous system (6.15) written for all projections. [21]

(ii) *Gluing via points.* $\hat{\mathtt{X}}_t$ is estimated as $\hat{\mathtt{X}}_t = \mathtt{V}_{1,2,3,4}{}^\top$. The fourth coordinates of points $\mathtt{X}$ estimated using (6.13) are not exactly one. Therefore, the closest vector from $\operatorname{span}\mathtt{X}^\top$ to vector $[1\,1\ldots1]^\top$ is found in the least squares as $\mathbf{v} = \mathtt{X}^\top(\mathtt{X}^\top)^+[1\,1\ldots1]^\top$ where $+$ stands for pseudoinverse. Then, $\mathbf{v}$ is replaced in $\operatorname{span}\mathtt{X}^\top$ by $[1\,1\ldots1]^\top$ as $\mathtt{X} := [\mathtt{U}(:,1:3), [1\,1\ldots1]^\top]^\top$ where $\mathtt{X}^\top - \mathbf{v}\mathbf{v}^+\mathtt{X}^\top = \mathtt{U}\operatorname{diag}(\sigma_1,\ldots,\sigma_z)\mathtt{V}^\top$ is the SVD factorization. Cameras are estimated as $\begin{bmatrix} \bar{\mathtt{P}}^i & \mathbf{t}^i \end{bmatrix} = (\mathtt{X}^\top_{\mathbf{p}^i})^+ \bar{\mathbf{x}}^i_{\mathbf{p}^i}$ where $\mathbf{p}^i$ denotes points observed by the i^{th} camera. Finally, points are estimated as $\bar{\mathbf{X}}_p = \bar{\mathtt{P}}^{i\,+}(\bar{\mathbf{x}}^{\mathbf{i}_p}_p - \mathbf{t}^i)$ where $\mathbf{i}_p$ denotes cameras observing the p^{th} point.

Results of gluing via cameras on open and closed Dinosaur sequence were 3.85 and 3.68 pxl, respectively. Results of gluing via points were better, see figure 6.4 and its caption. Metric upgrade was done using Guilbert's method [27] and his code downloadable from his web-page (see [27]). Focal length was set to 2000 as in Guilbert's code. Reprojection errors of the initial reconstruction on both open and closed sequences are below 2.7 pxl, which is twice lower than 5.4 pxl of the state-of-the art technique [27]. An improvement of this method with more experiments and comparison with the state of the art is reported in section 6.4.

Our technique using the affine model on a wider field of view resulted into mean reprojection errors of hundreds of pixels. In the St. George rotunda, a significant perspective effects are present as, for instance, cameras 22 and 65 are very close to the object, see figure 6.8right. Our conclusion is that this method can be used with the affine model for a narrow field of view only. It will be shown in the next section that the perspective model can be successfully used to model a wide field of view in this method.

6.3.5 Perspective Camera Model

In factorization using perspective camera model, if all the depths are close to equal, then an approximation to the reprojection error scaled by the common value of projective depths is minimized [31, *p*446]. Depths in an image pair can be estimated using method [113] from the epipolar geometry (EG). If the image pairs form a graph without cycles (tree), depths from individual image pairs can be easily chained and the result is known to be a set of depths consistent with all used EGs [113] even in case of missing data [17]. Nevertheless, in practical situations, many more EGs are available than the $m-1$ ones exploitable in an acyclic graph. Using overdetermined constraints on depths from all (reliable) EGs would naturally (i) result in better depth estimates and (ii) allow to relate data in image pairs within cycles, which concerns not only closed sequences but any wide baseline setup.

Cycles appear often in practice. For example in a closed sequence taken around an object, there is typically no point visible in all the images, as can be seen in figure 6.8. Although all subsequent cameras are close to each other in the graph of EGs, whatever tree is chosen, some cameras get always located at a large distance in the tree graph. Particularly, the larger is the amount of images of a scene available, the more cycles are likely to appear in the data.

[21] An improved version of gluing via affine cameras is in section 6.4, *p*70.

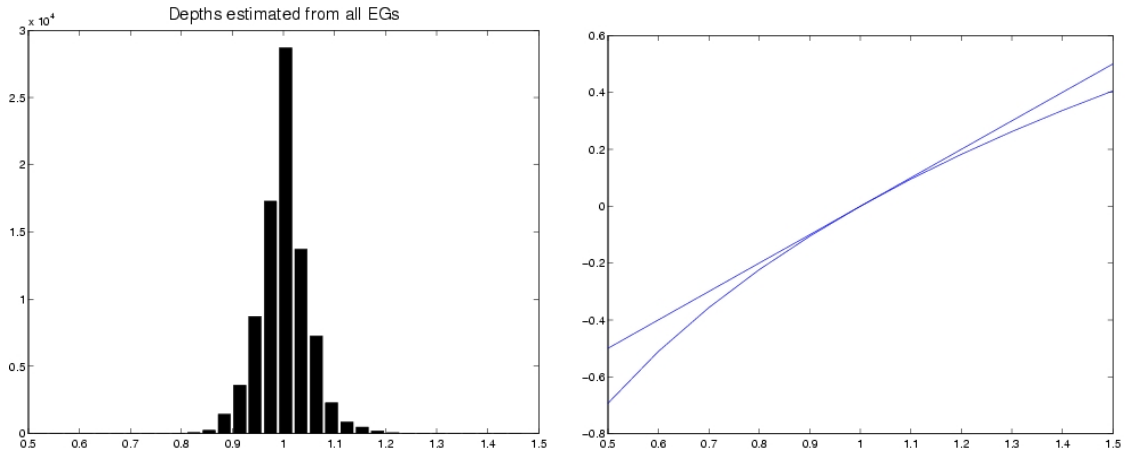


Figure 6.5: Depths estimated from all EGs: depths $\gamma_p^{ij,k}$ in the St. Martin rotunda balanced to be close to one (left) Change in scaling after applying logarithm: $\log x$ well approximates $x - 1$ (right).

Let $[\lambda_p^i \mathbf{x}_p^i]_{p\in\mathbf{p}_t}^{i\in\mathbf{1}_t} = \mathtt{U}\,\mathrm{diag}(\sigma_1,\ldots,\sigma_z)\mathtt{V}^\top$ be the SVD factorization. $\hat{\mathtt{P}}_t$ from (6.12) are estimated as $\hat{\mathtt{P}}_t = \mathtt{U}(:,1:4)$. $\hat{\mathbf{X}}_t$ from (6.13) are estimated as $\hat{\mathbf{X}}_t = \mathtt{V}(:,1:4)^\top$.

6.3.6 Overdetermined Depths

Consider EG between images i and j. Then, the corresponding image points can be scaled by $\gamma_p^{ij,k}$ as

$$\begin{bmatrix} \gamma_1^{ij,1}\mathbf{x}_{p_1}^i \ldots \gamma_z^{ij,1}\mathbf{x}_{p_z}^i \\ \gamma_1^{ij,2}\mathbf{x}_{p_1}^j \ldots \gamma_z^{ij,2}\mathbf{x}_{p_z}^j \end{bmatrix} = \begin{bmatrix} \hat{\mathtt{P}}^{ij,1} \\ \hat{\mathtt{P}}^{ij,2} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{X}}_{p_1} & \ldots & \hat{\mathbf{X}}_{p_z} \end{bmatrix} \tag{6.16}$$

where the right-hand side of the equation is the structure and motion in some projective frame. Depths in system (6.16) can be arbitrarily row- and column-wise rescaled [113]. However, whatever scaling is chosen, the connection to the scaling of the overall system of depths for all the data, λ_p^i, can be written as

$$\begin{bmatrix} r^{ij}[\lambda_{p_1}^i \ldots \lambda_{p_z}^i] \\ s^{ij}[\lambda_{p_1}^j \ldots \lambda_{p_z}^j] \end{bmatrix} = \begin{bmatrix} c_1^{ij}\begin{pmatrix}\gamma_1^{ij,1}\\ \gamma_1^{ij,2}\end{pmatrix} \ldots c_z^{ij}\begin{pmatrix}\gamma_z^{ij,1}\\ \gamma_z^{ij,2}\end{pmatrix} \end{bmatrix} \tag{6.17}$$

where r, s and c are some non-zero scalars defined for each image pair ij individually. Equation (6.17) relates all equivalent scalings corresponding to one class of projective reconstructions. System (6.17) consists of two by z equations. The c unknowns can be eliminated by dividing one row by the other:

$$r^{ij}/s^{ij} \begin{bmatrix} \frac{\lambda_{p_1}^i}{\lambda_{p_1}^j} & \frac{\lambda_{p_2}^i}{\lambda_{p_2}^j} & \ldots & \frac{\lambda_{p_z}^i}{\lambda_{p_z}^j} \end{bmatrix} = \begin{bmatrix} \frac{\gamma_1^{ij,1}}{\gamma_1^{ij,2}} & \ldots & \frac{\gamma_z^{ij,1}}{\gamma_z^{ij,2}} \end{bmatrix}.$$

After substituting unknowns r^{ij} and s^{ij} by $\alpha^{ij} = r^{ij}/s^{ij}$ and knowns γ's by $g_p^{ij} = \frac{\gamma_p^{ij,1}}{\gamma_p^{ij,2}}$, the equations can be rewritten as

$$\alpha^{ij} \begin{bmatrix} \lambda_{p_1}^i & \ldots & \lambda_{p_z}^i \end{bmatrix} = \begin{bmatrix} g_1^{ij}\lambda_{p_1}^j & \ldots & g_z^{ij}\lambda_{p_z}^j \end{bmatrix}. \tag{6.18}$$

These z equations are bilinear in unknowns α^{ij} and λ's. They can be "linearized" by applying logarithm to both sides of the equations, which is a reasonable operation because both α and λ's can be expected to be (i) positive due to oriented projective geometry (cheirality) [132, 31] and (ii) close to one, see figure 6.5a, where the log function well approximates function $x-1$, see figure 6.5b:

$$\begin{aligned}\log\alpha^{ij} + \begin{bmatrix} \log\lambda^i_{p_1} & \dots & \log\lambda^i_{p_z} \end{bmatrix} = \\ \begin{bmatrix} \log g^{ij}_1 + \log\lambda^j_{p_1} & \dots & \log g^{ij}_z + \log\lambda^j_{p_z} \end{bmatrix}.\end{aligned} \tag{6.19}$$

After substituting

$$\overline{\alpha} = \log\alpha, \overline{\lambda} = \log\lambda, \overline{g} = \log g, \tag{6.20}$$

(6.19) can be rewritten to

$$\overline{\alpha}^{ij} + \begin{bmatrix} \overline{\lambda}^i_{p_1} & \dots & \overline{\lambda}^i_{p_z} \end{bmatrix} = \begin{bmatrix} \overline{g}^{ij}_1 + \overline{\lambda}^j_{p_1} & \dots & \overline{g}^{ij}_z + \overline{\lambda}^j_{p_z} \end{bmatrix}.$$

Let all unknowns be rearranged to the left-hand side:

$$\begin{aligned}\overline{\alpha}^{ij} + \overline{\lambda}^i_{p_1} - \overline{\lambda}^j_{p_1} &= \overline{g}^{ij}_1 \\ &\vdots \\ \overline{\alpha}^{ij} + \overline{\lambda}^i_{p_z} - \overline{\lambda}^j_{p_z} &= \overline{g}^{ij}_z.\end{aligned} \tag{6.21}$$

After solving system (6.21), both $\overline{\lambda}$'s and $\overline{\alpha}$'s can be computed and back-substituted using (6.20). System (6.21) is sparse and hence can be solved efficiently by a sparse solver. The next section deals with the analysis of system (6.21) and ways of simplifying it by fixing some variables.

6.3.7 Freedom of Choice for Depths

Consider the matrix of all depths, $\Lambda = [\lambda^i_p]^{i\in1,\dots,m}_{p\in1,\dots,n}$. We will show that

Proposition 3 *Λ has freedom of choice for $m+n-1$ scales.*

Proof. As mentioned in [113], scales of the projection matrices $\mathtt{P}^i$, a^i, and scales of the 3D points $\mathbf{X}_p$, b_p, can be chosen arbitrarily, i.e. there is freedom of choice for $m+n$ numbers of vectors $\mathbf{a} = [a^1 \dots a^m]^\top$ and $\mathbf{b} = [b_1 \dots b_n]$. There is an equivalence relation on all pairs $\langle\mathbf{a},\mathbf{b}\rangle$ such that elements of the same class produce the same scaling, namely freedom up to an overall projective homography $h \in \mathbb{R}\backslash\{0\}$:

$$\langle\mathbf{a},\mathbf{b}\rangle \simeq \langle\hat{\mathbf{a}},\hat{\mathbf{b}}\rangle \equiv \exists h \in \mathbb{R}\backslash\{0\} : \begin{cases} \hat{a}^i &= a^i h \\ \hat{b}_p &= h^{-1} b_p \end{cases} \tag{6.22}$$

It can be easily seen that relation (6.22) is reflexive, symmetric and transitive, hence it is an equivalence. Depths λ^i_p correspond to a projective 3D reconstruction of the scene, $\{\mathtt{P}^i, \mathbf{X}_p\}$, in an arbitrary frame:

$$\lambda^i_p \mathbf{x}^i_p = \mathtt{P}^i\mathbf{X}_p = \mathtt{P}^i\mathtt{H}\mathtt{H}^{-1}\mathbf{X}_p.$$

where $\mathtt{H} \in \mathbb{R}^{4\times4}$, $|\mathtt{H}| \neq 0$. Note that depths and their scales are invariant to projective transformation of the scene $\mathtt{P} \mapsto \mathtt{P}\mathtt{H}$, $\mathtt{X} \mapsto \mathtt{H}^{-1}\mathtt{X}$. Changing scales of $\mathtt{P}^i$ and $\mathbf{X}_p$ appears as

$$\mathtt{P}^i a^i b_p \mathbf{X}_p = a^i b_p \lambda^i_p \mathbf{x}^i_p = \tilde{\lambda}^i_p \mathbf{x}^i_p \tag{6.23}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & * & * & * \\ 1 & * & * & * \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 & * & * \\ * & 1 & 1 & * \\ * & * & 1 & 1 \end{bmatrix}$$

Figure 6.6: Examples of fixing scales of Λ by fixing $m+n-1$ scale factors λ_p^i. Ones stand for fixed scale factors λ_p^i, stars stand for not fixed scales or the missing data.

For some $h \in \mathbb{R}\backslash\{0\}$,

$$\mathtt{P}^i a^i h h^{-1} b_p \mathbf{X}_p \quad = \quad \mathtt{P}^i \hat{a}^i \hat{b}_p \mathbf{X}_p = \hat{a}^i \hat{b}_p \lambda_p^i \mathbf{x}_p^i = \hat{\lambda}_p^i \mathbf{x}_p^i. \tag{6.24}$$

Because $a^i b_p = a^i h h^{-1} b_p$, the leftmost sides of equations (6.23) and (6.24) are equal and thus $\tilde{\lambda}_p^i = \hat{\lambda}_p^i$ which proves that elements of the same class of $\simeq$ produce the same scaling.

Although the $m+n$ numbers $\langle \mathbf{a}, \mathbf{b} \rangle$ can be chosen arbitrarily, due to existence of equivalence (6.22), Λ has freedom of choice for such number of scales that is lower by one, i.e. $m+n-1$. This applies to the case of the missing data as well. □

The $m+n-1$ scales capture the overall scale of Λ, $n-1$ ratios between column scales and $m-1$ ratios between row scales of Λ. We consider two practical ways of fixing scales of Λ:

1. By fixing $m+n-1$ scale factors λ_p^i chosen so that they fix all the row and column scale ratios. The ratio between two column scales can be fixed by fixing two depths in that columns which lie in a common row. Similarly for row scales. Examples can be seen in figure 6.6.

2. Alternatively, in context of equation (6.18), one can (i) fix the $m-1$ ratios between row scales by fixing the $m-1$ α's corresponding to some non-redundant set of EGs. Note that there may be more non-redundant sets of EGs. (ii) The n column scales can be fixed by fixing one arbitrary scale factor λ_p^i per column. Note that the ratios between column scales are fixed via the fixed row scale ratios using α's.

Note that in both ways the overall scale of Λ has been fixed by setting any scale factor λ_p^i.

Remark. There is a mistake in [113, section 2.1] according to which "once the $m+n$ overall scales ($\langle \mathbf{a}, \mathbf{b} \rangle$) have been fixed there is no further freedom of choice for the remaining $mn-m-n$ scale factors λ_p^i". The correct version is: "once the $m+n$ overall scales $\langle \mathbf{a}, \mathbf{b} \rangle$ have been fixed, the $m+n-1$ scale factors λ_p^i are fixed and there is no further freedom of choice for the remaining $mn-m-n+1$ scale factors λ_p^i". In case of the missing data there is less remaining scale factors λ_p^i.

1. Non-redundant set of EGs. Note that in this minimal case the $m-1$ EGs are organized in a tree graph. Each ij-EG brings n equations, each of which links projections of one point into images i and j. Hence, if no data are missing, system (6.21) consists of $(m-1)n$ equations with $mn+m-1$ unknowns (mn for $\overline{\lambda}_p^i$ and $m-1$ for $\overline{\alpha}^{ij}$). After fixing the $m+n-1$ unknowns ($m+n-1$ depths or n depths and $m-1$ α's), $mn-m-n+1+m-1 = (m-1)n$ unknowns remain. Thus, matrix $\mathtt{A}$ of system (6.21) is square with size $(m-1)n \times (m-1)n$. In case of the missing data, there is one unknown depth less for each missing image point, thus $\mathtt{A}$ is also square with size $k \times k$ where $k < (m-1)n$. If $\mathtt{A}$ is regular, existence of a unique solution to system (6.21) is guaranteed. Proof that matrix $\mathtt{A}$ is of full rank follows.

Proposition 4 *In case of non-redundant set of EGs, matrix* $\mathtt{A}$ *of system (6.21) is of full rank.*

Proof. For the two ways of fixing scales of Λ are equivalent, suppose without loss of generality that the $m-1$ α's have been fixed. Then, system (6.21) fragments into n independent systems

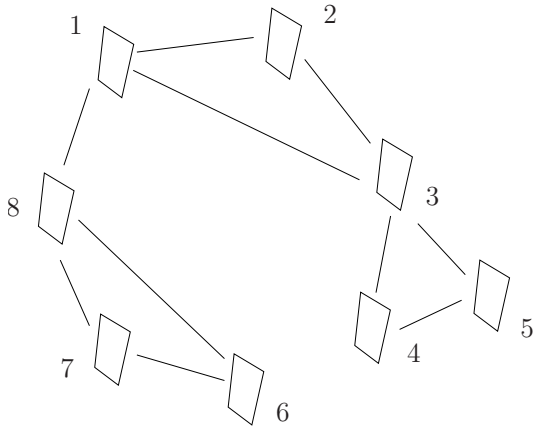


Figure 6.7: Redundant set of EGs: only the smallest redundant subgraphs (two-connected components) of EGs are needed to solve. Here, subgraphs 1-2-3, 3-4-5 and 6-7-8 can be solved independently.

of equations. Each subsystem links image points of just one 3D point because the only variables α's that could link equations corresponding to any two distinct 3D points have been eliminated. For each p, the p^{th} subsystem is of full rank: it consists of at most $m-1$ equations linking $\overline{\lambda}_p^1$, $\ldots$, $\overline{\lambda}_p^m$ in a tree graph which corresponds to the tree graph linking the $m-1$ EGs. Equations corresponding to edges in a tree graph are linearly independent as there are no cycles.

In case of the missing data, the tree graph linking the depths of the p^{th} point may fragment into smaller ones, each of which corresponds to independent equations again. When the track is fragmented, an arbitrary depth can be fixed in each fragment. □

Because system (6.21) can be fragmented into n independent systems of equations, each of these can be solved independently. Moreover, structure of these subsystems is trivial, thus, the equations can be recursively chained together to give estimates for the complete set of depths for point p [113], starting from the fixed depth and following the tree structure of the set of EGs.

2. Redundant set of EGs. In this case, system (6.21) consists of equations corresponding to some non-redundant set of EGs and some additional equations. So matrix $\mathtt{A}$ of system (6.21) is rectangular with size $k \times l$ where $k > l$, $l > s$ where s is the number of unknowns (and also equations) in the non-redundant system of equations. Note that all λ's have been introduced in the non-redundant system. Each of the additional equations either (i) introduces new variable α^{ij} and thus is independent or (ii) does not introduce any new variable and thus is dependent on the former equations in absence of noise. When no noise is present in the data, an exact solution to (6.21) exists. In presence of noise, an approximate solution is found in the sense depending on the used method. In our implementation, the quasi-minimal residual method (MATLAB's QMR) was used.

It turned out in our tests on real data that system (6.21) is sometimes fragmented into small subsystems which can be solved independently by simple recursive chaining. Even when the set of EGs is redundant as a whole, it may be fragmented into several two-connected components linked by trees. Each two-connected components can be solved independently, see figure 6.7. The data in images that are not included in any two-connected component can be discarded from system (6.21) because it is independent, thus it can be scaled afterwards. To obtain a unique reconstruction, several conditions have to be satisfied when the data fragments into more two-connected components.

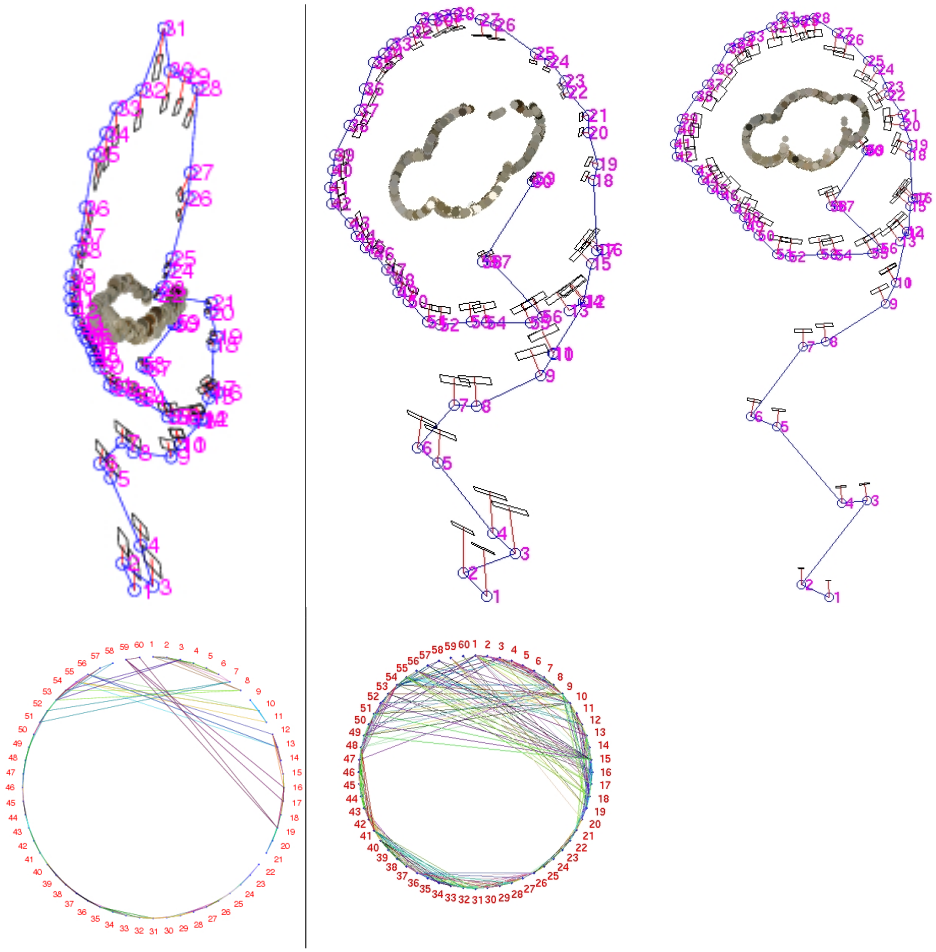


Figure 6.8: Reconstruction of the St. George rotunda captured on 60 images, 92% data missing: from a minimal set of 58 triplet constraints (left) from 166 triplets and after the metric BA (right).

Proposition 5 *The sufficient condition for a reconstruction of two two-connected components, A, B, sharing an articulation, c, ($c \in A \cap B$), to be unique is that there is at least one point, p, visible in camera triplet $\{a, b, c\}$, $a \in A$, $b \in B$, $a \neq c \neq b$ such that point p does not project into any epipole of any of epipolar geometries ac and bc.*

Proof. The only free parameter in the reconstruction of the two two-connected components is the scale between the two partial reconstructions. It can be easily estimated using one point satisfying the assumptions by rescaling one of the two partial reconstructions so that the distance between camera center c and the reconstruction of point p is same in both partial reconstructions.

Note that if point p is projected to some epipole in some of the two EGs, the scale would be undetermined in the corresponding partial reconstruction. This applies to the case when the three camera centers are collinear as well. □

6.3.8 Experiments

The MM of all scenes in this section except the Dinosaur sequence were obtained from pair-wise matches satisfying EGs between distinguished regions of various types detected in image pairs

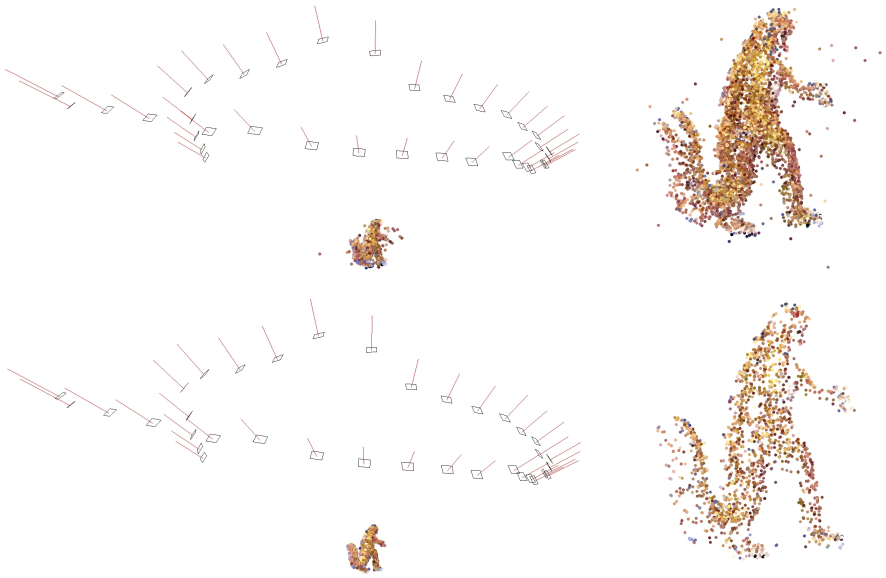


Figure 6.9: Metric reconstruction of the Dinosaur open sequence using the perspective camera model. Points reconstructed from the whole tracks (top) and from the tracks via images in the used image triplet constraints (bottom). The mean reprojection errors are 0.50 pxl (top) and 0.25 pxl (bottom). Note that the outlying points in (top) could prevent converging BA to the global minimum.

in a way similar to [17]. The threshold on distance to the epipolar lines was set to one pixel. All image triplet and image pair constraints with more than some given number of points were used. We tried also using only some of them with similar results. However, triplet constraints turned out to be necessary for reaching a sufficiently precise reconstruction for obtaining a reasonable metric upgrade. This demonstrates how much stronger are the constraints formed from view triplets compared to view pairs.

In this section, only results of the gluing via cameras are shown for the perspective model. It seems that gluing via points cannot be used in conjunction with the perspective model. At least we did not achieve any reasonable result using our implementation. Reconstructions from some minimal set of 58 (i.e. $m-2$) image triplet constraints and from 166 triplet constraints are shown in figure 6.8. In figure 6.8left, cameras 21 and 22 are reconstructed very far from each other compared to the surrounding cameras. This is because no constraint on camera pair 21 and 22 was used. Thus, it is clear that exploiting the cyclical structure of the data helps much in constraining the reconstruction. Recall, this would not be possible without depth consistency with EGs in a graph with cycles. Reconstruction using depths consistent only with EGs in an acyclic graph would look much worse than that in figure 6.8left.

An example of a wide baseline scene can be seen in figure 6.3. The St. Martin rotunda is very difficult to reconstruct because (i) both overview and detailed images are present (see top of figure 6.3a), (ii) some cameras are positioned very close to each other while some are very distant with wide baselines (see middle of figure 6.3a), (iii) it is a closed sequence around an object but at the same time there are many additional cycles (see bottom of figure 6.3a), making the task very challenging for sequential algorithms. The strong perspective effects make the task perhaps unsolvable for batch method [27] as it assumes affine cameras and slow motion.

6.3.9 Metric Reconstruction

Robust state-of-the art metric upgrade [78] was applied. However, if some cameras did not move along a fluent path with roughly the same distances between the consecutive frames, see the first eight cameras in figure 6.8right, the Nister's preconditioning based on this assumption could not provide a starting point sufficient for his optimization process to reach a good minimum. Thus, for non video-sequences, exhaustive search of the plane at infinity by sampling the space of its possible positions [33] was used instead. Even better results were achieved when exploiting the knowledge of ratios of focal lengths in the criterion function.

After the metric upgrade, most 3D points had the fourth coordinate positive. Only these were used in the metric bundle adjustment. Intrinsic parameters of all cameras were set to square pixel, principle point at image center and focal lengths to known ratios. The BA was done on a few points from each sampled submatrices $\mathbf{y}^{1_t}_{\mathbf{p}_t}$ with 3D points parameterized so that the fourth coordinate equals one. Because each bundled point was visible in two or three images only, there could be no outliers across many images (see figure 6.9) which could significantly obstruct converging close to the global minimum. Results of the metric BA can be seen in figures 6.3 and 6.8right.

This method provides a complete internal and external camera calibration and a sparse set of reconstructed points. Cameras can be used for dense reconstruction as in [17]. Figure 6.10 shows examples of disparity maps computed by method [52] on the Dinosaur and the St. George rotunda. The density approximation to point clouds, so called "fish-scales", shows that point clouds from individual image pairs fluently fade one into another thanks to correct gluing of partial reconstructions. For the Dinosaur, absence of any rough transition suggests reaching very close to the global minimum since we did not use the constraint that the sequence was closed but the result is a closed camera trajectory.

Solving (6.12) using MATLAB 6.5's EIGS took 0.25 seconds for 60 images of the St. George rotunda (PentiumIV@2.8GHz). Solving (6.21) using MATLAB's QMR took about one minute even for about 100 000 unknown projective depths. This time was reduced to seconds by sampling only a few points from each submatrix while achieving similar results.

Discussion and Conclusions

A method for fitting a low-rank matrix to a matrix with missing data was presented. Its correctness was demonstrated on an application to 3D reconstruction. In this approach, both affine and perspective camera models can be used. Affine model has the advantage of simplicity and stability if used on images taken by a distant camera. A linear method [27] is sufficient to get internal parameters close to the real ones to initiate the metric BA. On the other hand, the model gives high reprojection errors for a wider field of view. This does not happen when using the perspective camera model. Even very wide baseline scenes are reconstructed with reprojection errors around one pixel already by the linear method. Although using projective depths in the richer perspective model brings the necessity to estimate them, we showed that it is possible to estimate them reliably and consistently with all used EGs. Moreover, it has been shown that the richer perspective camera model does not overfit when used in our method.

There is a certain similarity between our method and Locally Linear Embedding (LLE) [98], although the tasks substantially differ. Our method is global in the same sense as LLE. Once the local structures (partial reconstructions) are chosen and fixed, they are combined by solving one optimization problem which has a global minimum as the eigenvalue problem is solved.

The perspective model can model omnidirectional cameras once points in omnidirectional images are attached to rays in space [73]. Other applications with missing data are possible, e.g., 3D reconstruction of non-rigid scenes. See more reconstructed scenes at [1].

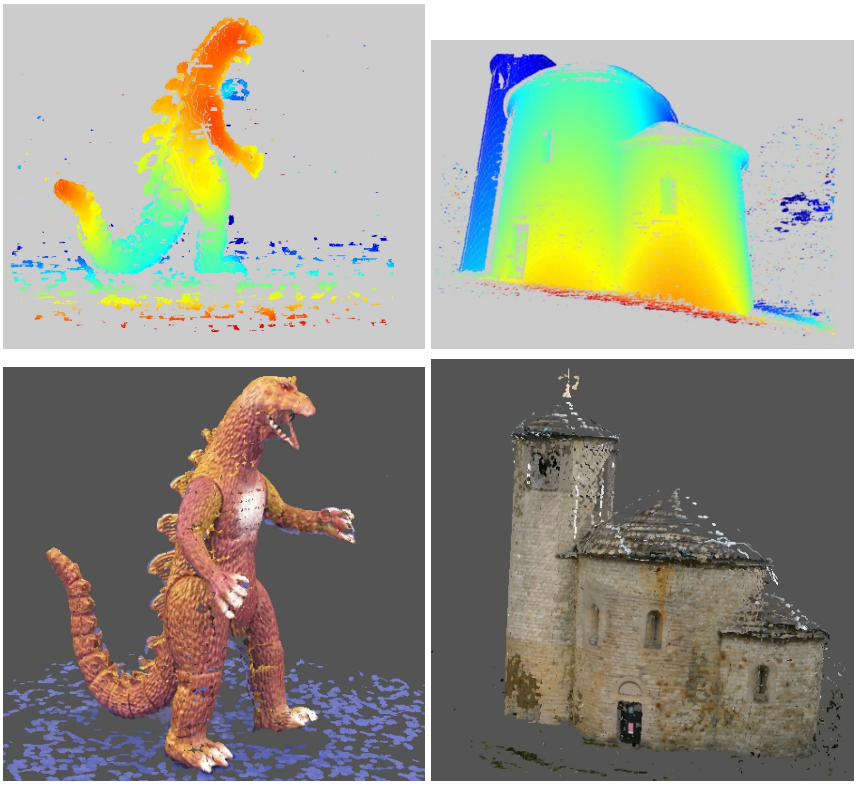


Figure 6.10: Application to dense matching (top) and dense reconstruction (bottom) on the Dinosaur sequence and the St. George rotunda: density of disparity map shows that the epipolar lines are correct.

Recently, a variation on the method described in this section appeared [118]. The main differences are that in method [118] (i) random sampling is used instead of factorizing full submatrices and (ii) center of gravity is subtracted from the data (the resulting matrix is called centered measurement matrix), reducing so the rank of the matrix. Note that the latter idea appears in the next section, which is a variation on the method presented in this section for affine cameras.

6.4 Gluing via Affine Cameras Revisited

In the paper by Buchanan & Fitzgibbon [11], 319 correct columns out of total 4983 columns in the Dinosaur sequence were manually selected. In this section, it will be shown that as simple method as [65] (section 6.3) can be used to obtain results comparable with the best results in [11]. In this section, only affine cameras are considered.

In section 6.3.4, the mean reprojection error on the Dinosaur open sequence using the algorithm "gluing via cameras" was 3.85 pxl, which is a worse result than the results reported in [11]. This happened due to several reasons:

1. This error concerns the affine-to-metric upgrade [27] and not the affine cameras.
2. The principle point was subtracted which is quite a good approximation of the center of gravity but not the best one w.r.t. the reprojection error.
3. Only a few submatrices containing data over three images were used.
4. Some outliers were present (those who survived RANSAC on the epipolar geometries of all image pairs).

After publishing [65], a small modification of gluing via affine cameras was done.

The New Modification

Recall gluing via affine cameras was proposed in section 6.3.4 [65]:

$$[\hat{\bar{\mathtt{P}}}_t \hat{\mathbf{t}}_t] \begin{bmatrix} \mathtt{A}_t & \mathbf{b}_t \\ \mathtt{0}_{1\times 3} & 1 \end{bmatrix} = [\bar{\mathtt{P}}^{\mathbf{i}_t} \mathbf{t}^{\mathbf{i}_t}] \quad \mathrm{t} = 1,\ldots,\mathrm{T}. \tag{6.25}$$

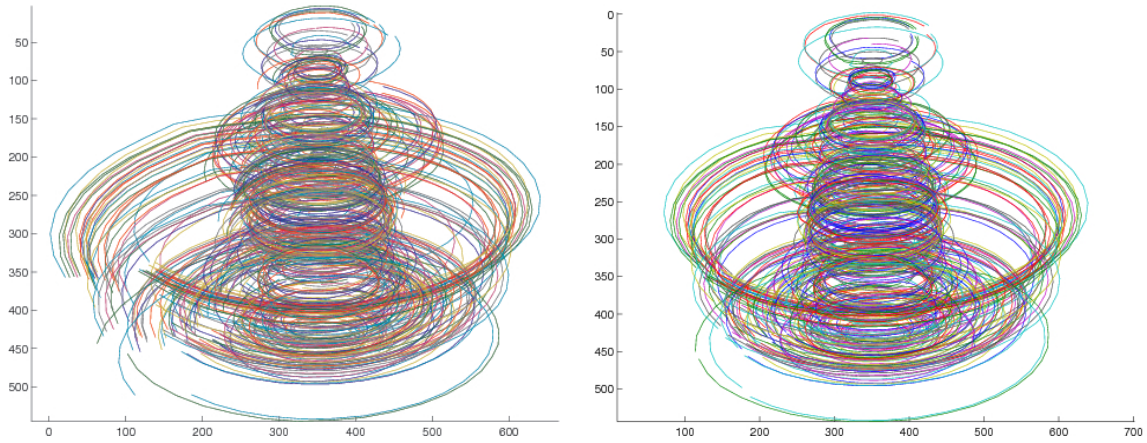
It turned out that it is better to subtract the center of gravity of each submatrix before its factorization. As in [65], first the consistent "affine" parts ($\mathtt{A}_t$ and $\bar{\mathtt{P}}$ in equation (6.25)) are estimated using $\hat{\bar{\mathtt{P}}}_t \mathtt{A}_t = \bar{\mathtt{P}}^{\mathbf{i}_t}$, followed by estimating consistent translations $\mathbf{t}$ and points $\bar{\mathbf{X}}_p$ using equation (6.15), *p*60.[22]

Experiments

1. When applied to the *outlier-free 319 points*, the mean reprojection error was only 1.03 pxl. Moreover, when using submatrices formed from the image points seen in all image pairs and triplets (686 submatrices in total), the mean reprojection error lowered to 0.9566 pxl, rms error to 1.3643. All reconstructed tracks are shown in figure 6.11 together with the result after affine bundle adjustment. Notice that perhaps the same result (rms error and tracks) is achieved as in [11, figure 3right] even without using priors on orthonormality of the camera matrices.

   The data contains some large residuals (our maximum residual is 40.3575) caused by the perspective effect present in the images, which is not captured by the affine camera model, and by mismatches. When minimizing the least squares error, the large residuals attract large attention and influence in the optimization. As a result, a random starting point

[22] This equation written for all points is a large non-homogenous system of equations. It was solved in [65] using MATLAB's QMR. We tried to convert the non-homogenous system of equations into a homogenous one by introducing a new variable for rescaling the right side and to solve it using MATLAB's EIGS. However, the results were very poor. Our possible explanation is that the one column of non-zeros (corresponding to the new variable for the right side) in the system matrix breaks its sparsity which is needed for EIGS to work properly.



mean/rms error 0.9566 / 1.3643 0.8810 / 1.2702

Figure 6.11: Chosen 319 reconstructed tracks of the Dinosaur sequence. The linear solution, see text (left) and after affine bundle adjustment (right) Notice that the same rms error is achieved as in [11, figure 3right] with tracks looking the same even without priors on orthonormality of the camera matrices.

may provide a solution with a lower rms error than ours such as 1.0847 pxl in [11, figure 3] without priors.

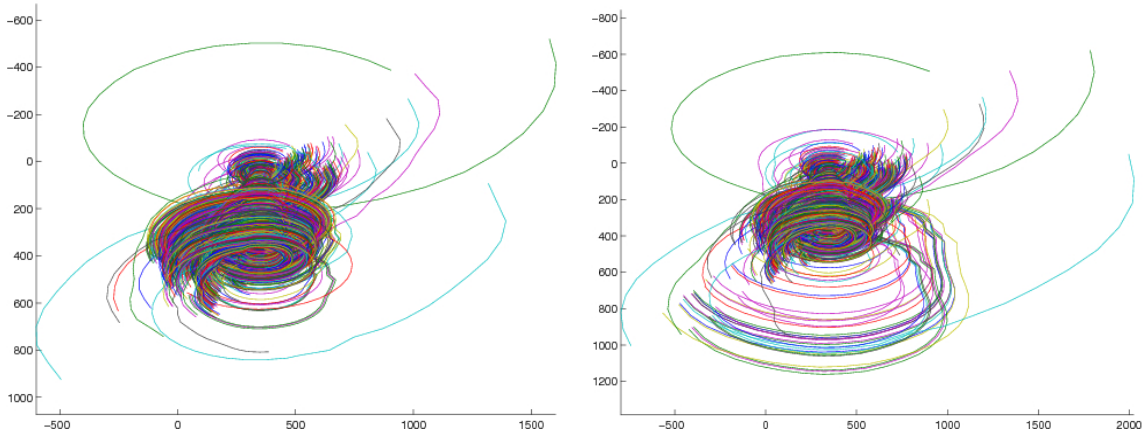
It turned out that at least four columns in the data matrix [11] contained mismatches. The columns with mismatches were removed in the following way. After removing the column with the largest reprojection error, the reconstruction was refined using bundle adjustment. The column with the largest error was manually checked if it contains correct data. If not, it was removed etc. After removing four columns (86, 89, 117, and 144), mean/rms errors of 0.6953 / 0.6800 pxl were achieved. Maximum residuum was 11.3277 pxl.

Our linear solution lead to correct solution (figure 6.11right) without doing many (tens of thousands) random initializations as in [11]. This clearly proves, that the difficulty paper [11] is trying to solve (missing data) does not exist when our version [65] (section 6.3) of Jacobs algorithm [43] is used, i.e. when minimizing an error in the measured entities instead of their subspace complements [65].

2. *All 4983 points* of the Dinosaur sequence with *no outlier removal*. Figure 6.12 shows the errors for 72 image triplets used in [65] and all image pairs and triplets. Even when the data contains mismatches, very similar solution to the outlier-free case (figure 6.11) is obtained.

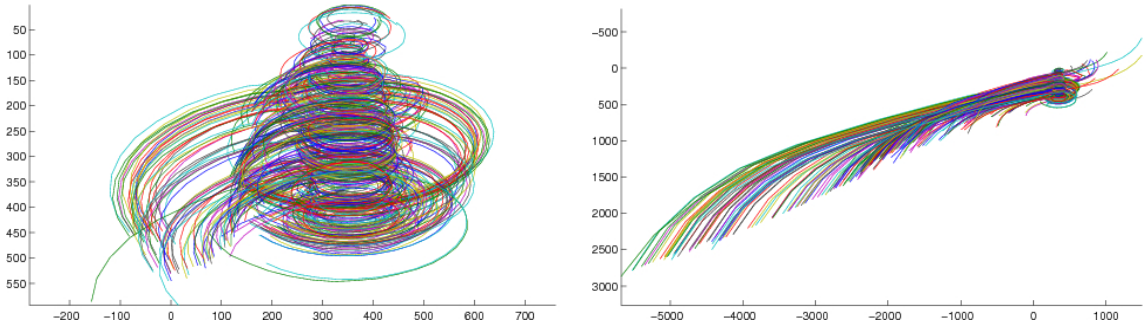
Remark

- One could try to estimate homographies including translations at the same time. It can be done so that all unknowns are searched for in the form of only one eigen vector of a matrix. However, in this parameterization, there is a coordinate ambiguity. The found reconstruction is determined up to an affine 3×3 transformation. One possible way to fix the coordinate frame is to set the first homography to $\mathtt{I}_{4\times4}$. To have homogenous system of equations, an auxiliary variable can be used instead of each occurrence of number one



mean/rms error 0.7735 / 1.3339 | 0.8524 / 1.4316

Figure 6.12: All 4983 reconstructed tracks of the Dinosaur sequence: 72 image triplets used in [65] (left) and all image pairs and triplets (right).



mean/rms error 0.9585 / 1.3516 | 1.3128 / 1.8305

Figure 6.13: All reconstructed tracks of the 319 points of the Dinosaur sequence: after 15 (left) and 36 (right) iterations of reestimating the translation subtracted instead of the center of gravity (see text).

in homographies.[23] Nevertheless, such a system has the disadvantage that the first three columns of the system, i.e. $\hat{\bar{\mathtt{P}}}_t \mathtt{A}_t = \bar{\mathtt{P}}^{\mathbf{i}_t}$, are independent of translations.[24]

Before the final bundle adjustment, one can try to improve the solution using the actual estimate of translation. It is subtracted from the submatrices instead of the center of gravity before factorizations. This can be iterated until convergence. As expected, the rms reprojection error decreases at first but after several (three) iterations, it increases again, oscillates, and after 15 iterations the solution clearly diverges, as can be seen in figure 6.13.

[23] It is clear now that the orthonormal matrices from factorization should not be reweighted by the corresponding singular values when searching for the solution as several eigen vectors of a matrix. On the contrary, when searching for the solution in the form of only one eigen vector, the data should be reweighted by the corresponding singular values (it was confirmed by experiments). In the first case, the reweighting is implicitly hidden in the eigenvalues being searched for. In the latter case, the reweighting must be done in the system matrix.

[24] In fact not, but to ensure the "affineness" of affine homographies, i.e. the three zeros in each of them, the translations, $\mathbf{t}^{\mathbf{i}_t}$, cannot influence anything except translations. Thus, "affine" parts can be estimated first and translations afterwards using all points, as it was done in section 6.3.4, *p*60.

Comments to Gluing via Points

In section 6.3.4, *p*60, a four-column matrix was used for representing points in partial reconstructions, $\hat{\mathbf{X}}_t$. In such a case, it makes no sence to use partial reconstructions over two images only, because the affine measurement (sub)matrix of two images has four rows and thus rank four. Hence (i) no information is removed in truncating the data matrix by choosing the four most significant vectors in factorization by SVD and (ii) in gluing partial reconstructions, all equations mapping such a partial reconstruction to the global one are trivially satisfied and thus do not constrain the global structure.

We tried to use the approach which proved to work for gluing via cameras, i.e. subtract the center of gravity of each submatrix prior to its factorization and use only the three most significant singular vectors for gluing via cameras. The vector of ones was attached to the structure at the end. However, much worse results were obtained.

6.5 Projective Gluing without Depth Consistency

This section proposes a new technique for 3D reconstruction from three views given only three pairwise epipolar geometries (EGs). No three-view correspondences are needed. The method searches jointly for the three cameras in the same projective frame and the three homographies mapping the three camera pairs (obtainable directly from the EGs) in different projective frames and with inconsistent scalings to a consistent camera triplet. Our solution minimizes a geometrically meaningful error which is close to the reprojection error. This enables (i) to cope well with noise and (ii) provides a stable technique even for collinear cameras in which case one of an infinite number of solutions is obtained. The second asset of this section is a new method for stitching partial reconstructions with inconsistent scalings of cameras. When combining with the above method for reconstructing view triplets, multiple-view reconstructions can be obtained while handling a large amount of occlusions. The technique was tested on both simulated and real image sequences in narrow and wide baseline settings. A number of images varying from three to hundred is reconstructed with reprojection errors below one pixel.

6.5.1 Three-view Reconstruction

The camera matrices corresponding to the fundamental matrix $\mathtt{F}$ representing an EG can be chosen up to a homography as

$$\begin{array}{rcll} \mathtt{P}^1 & = & [\mathtt{I} \,|\, \mathtt{0}] & \text{and} \\ \mathtt{P}^2 & = & [[\mathbf{e}']_\times \mathtt{F} \,|\, \mathbf{e}'] & \end{array} \tag{6.26}$$

where $\mathbf{e}'$ is the epipole satisfying $\mathbf{e}'^\top \mathtt{F} = 0$ [31, p256]. We will solve the following problem.

Task 1 *Given three EGs between images 1-2, 2-3, and 1-3, estimate camera matrices 1, 2, and 3 satisfying the EGs.*

There exists a method [31, p386] which computes the third projection matrix $\mathtt{P}^3$ using equations based on skew-symmetricity of ${\mathtt{P}^3}^\top \mathtt{F}_{31} \mathtt{P}^1$ and ${\mathtt{P}^3}^\top \mathtt{F}_{32} \mathtt{P}^2$. Each of these matrices gives rise to 10 linear equations in the entries of $\mathtt{P}^3$, altogether 20 equations in the 12 entries of $\mathtt{P}^3$. From these, $\mathtt{P}^3$ may be computed linearly. However, the error minimized has no well motivated connection to image measurements.

Therefore, this method can solve the task under two restrictive assumptions: (i) the camera centers are not collinear and (ii) the three fundamental matrices $\mathtt{F}_{21}$, $\mathtt{F}_{32}$, $\mathtt{F}_{31}$ are compatible, i.e.

$$\mathbf{e}_{23}^\top \mathtt{F}_{21} \mathbf{e}_{13} = \mathbf{e}_{31}^\top \mathtt{F}_{32} \mathbf{e}_{21} = \mathbf{e}_{32}^\top \mathtt{F}_{31} \mathbf{e}_{12} = 0.$$

where $\mathbf{e}_{\bullet\bullet}$ are epipoles. If data are noisy, the fundamental matrices are practically never compatible. Moreover, in sequences, consecutive cameras are often almost collinear.

In [107], an asymmetric approach was proposed to achieve compatible fundamental matrices. Two epipolar geometries were taken for granted and the third one was projected down on the 4D linear space to which the two other ones would constrain it. This technique is simpler than [31, p386] and works quite well in practice.

We present a method that minimizes a geometrically meaningful error, more specifically a reasonable approximation of the reprojection error in the images. Moreover, in contrast to method [31, p386], our reconstruction is stable even for collinear cameras (in which case one of an infinite number of solutions is obtained) and in contrast to [107], it is symmetric.

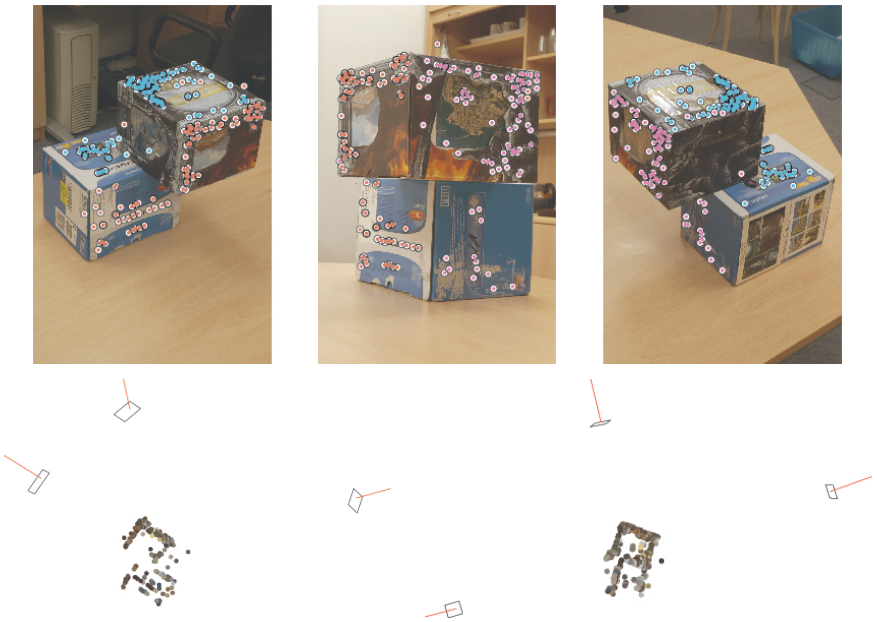


Figure 6.14: An extreme case of the missing data: image correspondences shown in three colors corresponding to three image pairs. There is no point visible in three images (top) Reconstruction of the boxes from two view points (bottom).

The Main Idea

The main idea of the new algorithm is to align the pairwise reconstructions using homographies while compensating for inconsistencies in scaling of projection matrices. Let the camera pairs corresponding to the 1-2, 2-3, and 1-3 EGs be denoted as $\begin{bmatrix} \mathtt{P}^1_{12} \\ \mathtt{P}^2_{12} \end{bmatrix}$, $\begin{bmatrix} \mathtt{P}^2_{23} \\ \mathtt{P}^3_{23} \end{bmatrix}$, and $\begin{bmatrix} \mathtt{P}^1_{13} \\ \mathtt{P}^3_{13} \end{bmatrix}$, respectively, where each $\mathtt{P}^{\bullet}_{\bullet}$ stands for a 3×4 matrix. It is possible to estimate 4×4 homographies $\mathtt{H}_2$ and $\mathtt{H}_3$ mapping coordinate systems of camera pairs 2-3 and 1-3, respectively, to coordinates of camera pair 1-2 using the following system of equations:

$$\left.\begin{aligned} \begin{bmatrix} s\,\mathtt{P}^2_{12} \\ \mathtt{P}^3 \end{bmatrix} &= \begin{bmatrix} \mathtt{P}^2_{23} \\ \mathtt{P}^3_{23} \end{bmatrix} \mathtt{H}_2 \\ \begin{bmatrix} r\,\mathtt{P}^1_{12} \\ \mathtt{P}^3 \end{bmatrix} &= \begin{bmatrix} \mathtt{P}^1_{13} \\ \mathtt{P}^3_{13} \end{bmatrix} \mathtt{H}_3 \end{aligned}\right\} \tag{6.27}$$

where s and r are nonzero scales of projection matrices and $\mathtt{P}^3$ stands for the projection matrix of camera 3 in the coordinate system of camera pair 1-2.

System (6.27) is linear in all unknowns s, $\mathtt{H}_2$, $\mathtt{H}_3$ and $\mathtt{P}^3$. It consists of $4 \cdot 12 = 48$ equations and $2 + 2 \cdot 16 + 12 = 46$ unknowns. Hence, system (6.27) is slightly overconstrained. System (6.27) contains 46 linearly independent equations iff the three EGs correspond to three cameras with distinct centers. Note that the camera centers may be collinear.

Signs of r and s are related to the choice of bases of coordinate systems in cameras. If all bases have the same handedness, i.e. they are all left-handed or all right-handed, and represent equally oriented coordinate systems, then $\operatorname{sign}(r) = \operatorname{sign}(s)$.

If signs of r and s differ, sign of some camera in one camera pair w.r.t. the remaining two camera pairs is wrong. It can be easily repaired by switching the sign of an arbitrary camera in any of the three pairs. Then, homographies $\mathtt{H}_2$ and $\mathtt{H}_3$ have the same signs of their determinants, i.e. they are both orientation preserving or both orientation reversing.

Orientation of cameras $\mathtt{P}^1$, $\mathtt{P}^2$ can be estimated using the oriented projective geometry (OPG) by exploiting also image measurements [131, 132]. However, it is not necessary to find correct camera orientations in advance as they can be easily recovered from the sings of r and s in (6.27).

$\mathtt{P}^3$ can be eliminated using the $2^{\mathbf{nd}}$ row in (6.27):

$$\left.\begin{array}{rrcr} s\,\mathtt{P}^2_{12} & & = & \mathtt{P}^2_{23}\mathtt{H}_2 \\ r\,\mathtt{P}^1_{12} & & = & \mathtt{P}^1_{13}\mathtt{H}_3 \\ & \mathtt{P}^3_{23}\mathtt{H}_2 & = & \mathtt{P}^3_{13}\mathtt{H}_3 \end{array}\right\} \tag{6.28}$$

If noise is present, it may happen that magnitudes of s and r are significantly lower and the solution is unstable which may lead to that signs of s and r become different even when camera orientations in $\mathtt{P}^{\bullet}_{\bullet}$ are correct. This occurs when such solution of (6.28) is found for which $\|\mathtt{P}^1_{13}\mathtt{H}_3\| << \|\mathtt{P}^3_{13}\mathtt{H}_3\|$ or $\|\mathtt{P}^2_{23}\mathtt{H}_2\| << \|\mathtt{P}^3_{23}\mathtt{H}_2\|$ where $\|\cdot\|$ denotes the Frobenius norm. Such situation may occur because system (6.28) is overparameterized. Fixing one of the scales s or r to one prevents the instability. Supposing r is fixed to one, the following non-homogenous system is obtained: [25]

$$\left.\begin{array}{rrrcr} s\,\mathtt{P}^2_{12} & -\mathtt{P}^2_{23}\mathtt{H}_2 & & = & \mathtt{0}_{3\times 4} \\ & & \mathtt{P}^1_{13}\mathtt{H}_3 & = & \mathtt{P}^1_{12} \\ & -\mathtt{P}^3_{23}\mathtt{H}_2 & +\mathtt{P}^3_{13}\mathtt{H}_3 & = & \mathtt{0}_{3\times 4} \end{array}\right\} \tag{6.29}$$

It is possible to fix one camera per camera pair to $[\mathtt{I}|\mathtt{0}]$:

$$\begin{bmatrix} \mathtt{P}^1_{12} \\ \mathtt{P}^2_{12} \end{bmatrix} = \begin{bmatrix} \mathtt{I}|\mathtt{0} \\ \mathtt{A}|\mathbf{a} \end{bmatrix}, \quad \begin{bmatrix} \mathtt{P}^2_{23} \\ \mathtt{P}^3_{23} \end{bmatrix} = \begin{bmatrix} \mathtt{B}|\mathbf{b} \\ \mathtt{I}|\mathtt{0} \end{bmatrix}, \quad \begin{bmatrix} \mathtt{P}^1_{13} \\ \mathtt{P}^3_{13} \end{bmatrix} = \begin{bmatrix} \mathtt{I}|\mathtt{0} \\ \mathtt{C}|\mathbf{c} \end{bmatrix}. \tag{6.30}$$

Then, from the second row in (6.29),

$$\mathtt{H}_3 = \begin{bmatrix} \mathtt{I}_{3\times 3} \\ \mathbf{h}^\top \end{bmatrix} \tag{6.31}$$

for some $\mathbf{h} \in \mathbb{R}^{4\times 1}$. After denoting

$$\mathtt{H}_2 = \begin{bmatrix} \mathtt{G}_{3\times 3} \\ \mathbf{g}^\top \end{bmatrix} \tag{6.32}$$

where $\mathbf{g} \in \mathbb{R}^{4\times 1}$, the third row in (6.29) becomes:

$$-[\mathtt{I}|\mathtt{0}]\begin{bmatrix} \mathtt{G} \\ \mathbf{g}^\top \end{bmatrix} + [\mathtt{C}|\mathbf{c}]\begin{bmatrix} \mathtt{I} \\ \mathbf{h}^\top \end{bmatrix} = \mathtt{0}_{3\times 4}$$

from which

$$\mathtt{G} = [\mathtt{C}|\mathtt{0}] + \mathbf{c}\mathbf{h}^\top. \tag{6.33}$$

The first row in (6.29) becomes after substituting for $\mathtt{G}$ from (6.33)

$$s[\mathtt{A}|\mathbf{a}] \quad -[\mathtt{B}|\mathbf{b}]\begin{bmatrix} [\mathtt{C}|\mathtt{0}] + \mathbf{c}\mathbf{h}^\top \\ \mathbf{g}^\top \end{bmatrix} = \mathtt{0}_{3\times 4}$$

[25] Scale of $\mathtt{H}_3$ is chosen by the second equation in (6.29). Consequently, scale of $\mathtt{H}_2$ is chosen relatively to scale of $\mathtt{H}_3$ by the last equation in (6.29). Freedom for "breathing" of scale of $\mathtt{H}_2$ (and thus also scale of $\mathtt{H}_3$) is ensured by varying s, which scales $\mathtt{P}^2_{12}$ in the first equation in (6.29).

and after some manipulations one obtains

$$s\ [\mathtt{A}|\mathbf{a}] \quad -\mathtt{B}\mathbf{c}\mathbf{h}^\top - \mathbf{b}\mathbf{g}^\top \quad = \quad [\mathtt{BC}|\mathtt{0}]. \tag{6.34}$$

System (6.34) has twelve equations for nine unknowns s, $\mathbf{g}$, and $\mathbf{h}$. Residua in cameras one and three are guaranteed to be zero due to substitutions (6.31) and (6.33). Thus, in system (6.34), residua are minimized only in camera two.[26]

We observed in our experiments that (6.34) produced mostly about 5% lower residua in comparison with (6.29) expressed in Frobenius norm. As the error minimized in (6.34) is spread over one camera only, the resulting reconstructions can be expected to be less "consistent" with reconstructions of the overlapping camera triplets. This was also confirmed by our experiments. The whole Dinosaur sequence was reconstructed with almost the same mean reprojection errors 0.24 and 0.25 pxl using systems (6.29) and (6.34), respectively. Triplets over at most five images 1-2-3, 2-3-4, 3-4-5, 1-3-5, 4-5-6, 2-4-6, etc. were used.

Goldberger [25] allows only four or five parameters per homography to vary in solving the three-view reconstruction from EGs, compare to our sixteen in (6.29). Therefore, his solution is suboptimal as well as our solution (6.34). Goldberger's solution for more than three views can solve more difficult scenes than our solution, like the cube configuration. However, for the scenes which we are able to reconstruct, our solution should give better results, because we compute the scales using overdetermined system, unlike Goldberger who uses a minimal subset of all constraints on the scales. It is an interesting question if it is possible to extend his solution for the overdetermined constraints. While we minimize the quality (error) of alignment of pairs into the consistent camera set, Goldberger minimizes (once having estimated scales β_{ij}) the consistency of each consistent camera pair with $\mathtt{F}$, see [25, theorem 1]. Note that this is not equivalent.

As noted above, spreading the minimized error just over one of the two cameras in each camera pair leads to worse results. However, there is even a more serious problem of unsymmetricity related to a special role of the third projection matrix $\mathtt{P}^3$ with respect to the remaining projection matrices. In method [31, *p*386] and in system (6.27), projection matrix $\mathtt{P}^3$ is searched for while having fixed the other two projection matrices $\mathtt{P}^1$ and $\mathtt{P}^2$. In systems (6.29) and (6.34), although $\mathtt{P}^3$ is eliminated, homographies from only two (of three) camera pairs are estimated.

In the unsymmetric formulation one makes a choice of some parameters which are to be fixed. Thus, there is a smaller parameter space over which the optimization problem is minimized, compared to a symmetric formulation. For the fixed parameters cannot compensate for the errors in measurements (EGs), the unsymmetric formulation produces in general worse solutions than the symmetric one. A symmetric solution given in the next section involves estimation of all three homographies.

Symmetric Solution

The symmetric solution comes out of system (6.12), *p*59, where consistent scalings of cameras was achieved by using points seen across three of more views. The assumption of consistent camera scalings is not valid in our task but we will show how to use method [65] (section 6.3) for the case of inconsistent scalings. In the situation with three camera pairs 1-2, 2-3 and 1-3,

[26] System (6.34) can be solved efficiently without any matrix inversion. After partitioning equations $\mathtt{A}^\top\mathtt{A} = \mathtt{A}^\top\mathbf{b}$ in the way as equations (A6.6) in [31, *p*604], only inverse of a real number is needed instead of inverse of a 9×9 matrix.

system (6.12), *p*59 becomes using the notation introduced here

$$\left.\begin{array}{rcl} \omega_1 \left(\begin{bmatrix} \mathtt{P}^1 \\ \mathtt{P}^2 \end{bmatrix} - \begin{bmatrix} \mathtt{P}^1_{12} \\ \mathtt{P}^2_{12} \end{bmatrix} \mathtt{H}_1 \right) & = & \mathtt{0}_{6\times 4} \\ \omega_2 \left(\begin{bmatrix} \mathtt{P}^2 \\ \mathtt{P}^3 \end{bmatrix} - \begin{bmatrix} \mathtt{P}^2_{23} \\ \mathtt{P}^3_{23} \end{bmatrix} \mathtt{H}_2 \right) & = & \mathtt{0}_{6\times 4} \\ \omega_3 \left(\begin{bmatrix} \mathtt{P}^1 \\ \mathtt{P}^3 \end{bmatrix} - \begin{bmatrix} \mathtt{P}^1_{13} \\ \mathtt{P}^3_{13} \end{bmatrix} \mathtt{H}_3 \right) & = & \mathtt{0}_{6\times 4} \end{array}\right\} \tag{6.35}$$

where ω_t denotes the weight of the t^{th} EG taking into consideration the belief of the EG estimate expressed in terms of the number of matches n_t [27] satisfying the EG:

$$\omega_t = \sqrt{\frac{n_t}{\bar{n}}}$$

where $\bar{n}$ is the average number of points in EGs. Normalization by $\bar{n}$ gets all weights close to one.

In system (6.35), all unknown cameras $\mathtt{P}^1$, $\mathtt{P}^2$ and $\mathtt{P}^3$ in the same projective frame and all the three homographies mapping the three camera pairs in different projective frames are estimated. It was shown in section 6.3.2, *p*58, that system (6.35) (which is a variation on systems (6.12), *p*59 and (6.6), *p*56) minimizes a reasonable approximation of the reprojection error. A very good behaviour was demonstrated in [65] (section 6.3) on reconstructions of difficult scenes of various types captured in many images in narrow and wide baseline setups. The same error is minimized in all equations introduced in section 6.5.1. Nevertheless, there is a substantial difference between systems from section 6.5.1 and system (6.35): the latter is symmetric w.r.t. all views, thus a better resistance to noise can be expected, as will be shown in experiments.

Consistent scales of camera matrices cannot be expected if no three-view correspondences are present in the data. Therefore, the correct equations mapping the camera pairs to the cameras in the same projective frame must take into account possible inconsistencies among scales of the projection matrices. The scale inconsistencies can be expressed using one scale factor, s, only, similarly as in system (6.29):

$$\left.\begin{array}{rcl} \omega_1 \left(\begin{bmatrix} s\,\mathtt{P}^1 \\ \mathtt{P}^2 \end{bmatrix} - \begin{bmatrix} \mathtt{P}^1_{12} \\ \mathtt{P}^2_{12} \end{bmatrix} \mathtt{H}_1 \right) & = & \mathtt{0}_{6\times 4} \\ \omega_2 \left(\begin{bmatrix} \mathtt{P}^2 \\ \mathtt{P}^3 \end{bmatrix} - \begin{bmatrix} \mathtt{P}^2_{23} \\ \mathtt{P}^3_{23} \end{bmatrix} \mathtt{H}_2 \right) & = & \mathtt{0}_{6\times 4} \\ \omega_3 \left(\begin{bmatrix} \mathtt{P}^1 \\ \mathtt{P}^3 \end{bmatrix} - \begin{bmatrix} \mathtt{P}^1_{13} \\ \mathtt{P}^3_{13} \end{bmatrix} \mathtt{H}_3 \right) & = & \mathtt{0}_{6\times 4}. \end{array}\right\} \tag{6.36}$$

By introducing scale factor s, the first equation in (6.36) became bilinear. Fortunately, linearity can be achieved by fixing the projective ambiguity of the unknown cameras $\mathtt{P}^1$, $\mathtt{P}^2$ and $\mathtt{P}^3$. It can be done for instance by setting the first four rows of $\begin{bmatrix} \mathtt{P}^1 \\ \mathtt{P}^2 \\ \mathtt{P}^3 \end{bmatrix}$ to $-\mathtt{I}_{4\times 4}$: $\begin{bmatrix} \mathtt{P}^1 \\ \mathtt{P}^2 \\ \mathtt{P}^3 \end{bmatrix} = \begin{bmatrix} -\mathtt{I}_{4\times 4} \\ \mathtt{B} \\ \mathtt{C} \end{bmatrix}$ where matrices $\mathtt{B}$ and $\mathtt{C}$ have sizes 2×4 and 3×4, respectively. Then, system (6.36)

[27] This definition of belief is quite limited as not only the number of correspondences is important but also what vectors represent them. For instance, thousand correspondences close to an epipole may provide a weaker constraint on the EG compared to a few points evenly spread on the view-sphere.

becomes

$$\left.\begin{array}{rl}
\omega_1 \left(\begin{bmatrix} -s & 0 & 0 & 0 \\ 0 & -s & 0 & 0 \\ 0 & 0 & -s & 0 \\ 0 & 0 & 0 & 0 \\ & \mathtt{B} & & \end{bmatrix} - \begin{bmatrix} \mathtt{P}_{12}^1 \\ \mathtt{P}_{12}^2 \end{bmatrix} \mathtt{H}_1 \right) & = \omega_1 \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ & \mathtt{0}_{2\times 4} & & \end{bmatrix} \\
\omega_2 \left(\begin{bmatrix} 0 & 0 & 0 & 0 \\ & \mathtt{B} & & \\ & \mathtt{C} & & \end{bmatrix} - \begin{bmatrix} \mathtt{P}_{23}^2 \\ \mathtt{P}_{23}^3 \end{bmatrix} \mathtt{H}_2 \right) & = \omega_2 \begin{bmatrix} 0 & 0 & 0 & 1 \\ & \mathtt{0}_{2\times 4} & & \\ & \mathtt{0}_{3\times 4} & & \end{bmatrix} \\
\omega_3 \left(\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ & \mathtt{C} & & \end{bmatrix} - \begin{bmatrix} \mathtt{P}_{13}^1 \\ \mathtt{P}_{13}^3 \end{bmatrix} \mathtt{H}_3 \right) & = \omega_3 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ & \mathtt{0}_{3\times 4} & & \end{bmatrix}
\end{array}\right\} \tag{6.37}$$

In system (6.37), the scale factor between camera scales is estimated jointly with the three consistent camera matrices (whose some parameters have been fixed) and the three homographies. Thus, $1{+}8{+}12{+}3{\cdot}16 = 69$ unknowns are solved for using $3{\cdot}24 = 72$ equations. Note that fixing the 16 degrees of freedom of the projective frame by setting some 16 camera parameters does not harm the symmetricity w.r.t. all cameras as it has no effect on the minimized error.[28]

In real scenes in this work, the two-view matches satisfying EGs were found as distinguished regions of various types detected in image pairs [69]. Results of a three-view reconstruction using two-view correspondences only can be seen in figure 6.14. The initial projective reconstruction was obtained with the mean reprojection error 0.62 pxl. In this work, metric upgrade was done using [78]. The error after the metric bundle adjustment was 0.66 pxl.

6.5.2 Gluing Unscaled Reconstructions

Suppose some triplets of views have been reconstructed, either using the above described method exploiting only EGs or any other method. If the corresponding camera matrices in triplets had consistent scalings, method for gluing partial reconstructions [65] (section 6.3) could be used. In [65], consistent scaling of cameras was ensured using image points via their depths. Note that this cannot be done if no three-view correspondences are present. In this section we will show how to glue reconstructions with inconsistent scalings of camera matrices.

Suppose two view triplets have a two-view overlap in a camera pair. (This does not necessarily mean that there are three-view correspondences.) Two reconstructions $\begin{bmatrix} \mathtt{A} \\ \mathtt{B} \end{bmatrix}$ and $\begin{bmatrix} \mathtt{C} \\ \mathtt{D} \end{bmatrix}$ of the camera pair with inconsistent scalings of projection matrices $\mathtt{A}$, $\mathtt{B}$, $\mathtt{C}$, and $\mathtt{D}$ can be aligned using the following equations:

$$\left.\begin{bmatrix} s\,\mathtt{A} \\ r\,\mathtt{B} \end{bmatrix} = \begin{bmatrix} \mathtt{C} \\ \mathtt{D} \end{bmatrix} \mathtt{H}\right\} \tag{6.38}$$

Similarly to system (6.28), we set, e.g., s to one. A smaller non-homogenous system of equations is obtained which can also be solved slightly faster:

$$\left.\begin{array}{rcl} \mathtt{CH} & = & \mathtt{A} \\ -r\,\mathtt{B} \quad +\mathtt{DH} & = & \mathtt{0}_{3\times 4} \end{array}\right\} \tag{6.39}$$

We observed a neglecting difference in numerical behaviour between systems (6.38) and (6.39). If scale r turns out to be negative, camera scalings are not consistent w.r.t. the oriented projective

[28] System (6.37) can be simplified by eliminating the camera unknowns, $\mathtt{B}$ and $\mathtt{C}$, using [31, *p*604].

geometry *(OPG)* [131, 132]. As well as is section 6.5.1, reversing the sign of an arbitrary projection matrix ensures the consistency. [29]

Once having estimated scale r and homography $\mathtt{H}$, the two camera triplets can be glued using them.

6.5.3 Gluing Many Unscaled Reconstructions

If multiple triplets are to be glued, all their two-view overlaps form an overdetermined set of constraints on the camera scales in practical situations. In this work, only three-view reconstructions are glued. Nevertheless, the method described below can be used for gluing larger reconstructions with inconsistent scalings.

Each camera in each triplet can be rescaled by some scale factor to achieve consistency with other cameras. Let the scale factor be denoted as s_t^i where i stands for the camera index and t stands for the triplet index. For each two-view overlap in images i and j of some two triplets u and v, system (6.39) gives the following constraint on scales:

$$\frac{s_u^i}{r^{ijuv} s_u^j} = \frac{s_v^i}{s_v^j} \tag{6.40}$$

where $r^\bullet$ has been estimated using system (6.39). In fact, unknowns $s_\bullet^\bullet$ in equation (6.40) are quadrilinear:

$$\frac{s_u^i s_v^j}{s_u^j s_v^i} = r^{ijuv} \tag{6.41}$$

This can be easily linearized by applying logarithm to both sides of the equation which is correct under the assumption that the terms to which it is applied are positive. The positiveness can be easily ensured by the technique for sign switching described above. Then, equation (6.41) becomes

$$\bar{s}_u^i + \bar{s}_v^j - \bar{s}_u^j - \bar{s}_v^i = \bar{r}^{ijuv} \tag{6.42}$$

where $\bar{s}_u^i = \log s_u^i$, $\bar{s}_u^j = \log s_u^j$, $\bar{s}_v^i = \log s_v^i$, $\bar{s}_v^j = \log s_v^j$, and $\bar{r}^{ijuv} = \log r^{ijuv}$.

A system consisting of equations of form (6.42) is sparse (four non-zeroes per row), and thus can be efficiently solved, e.g., by MATLAB's QMR. To reduce the number of unknowns, one camera per each triplet can be fixed because what matters are mutual ratios of scales within a triplet.

When using our method described in section 6.5.1, equations (6.39) do not have to be used at all because the needed ratio between the two camera scales in the two-view overlap can be obtained directly from scale factors s estimated using system (6.37):

$$\bar{s}_u^i + \bar{s}_v^j - \bar{s}_u^j - \bar{s}_v^i = \log \frac{c_u^i c_v^j}{c_u^j c_v^i}. \tag{6.43}$$

The c scales in (6.43) are computed so that for each triplet t relating images i, j and k, scales $c_t^\bullet$ are set to

$$\begin{bmatrix} c_t^i \\ c_t^j \\ c_t^k \end{bmatrix} = \begin{bmatrix} s \\ 1 \\ 1 \end{bmatrix}$$

[29] After partitioning equations $\mathtt{A}^\top \mathtt{A} = \mathtt{A}^\top \mathbf{b}$ in the way as equations (A6.6) in [31], only one inverse of a symmetric positive-definite 4×4 matrix is needed instead of inverse of a 17×17 matrix.

Further, after eliminating all $\mathtt{H}$ parameters, much faster estimation can be expected. This was not tried.

where scale factor s was estimated using system (6.37).

The Dinosaur sequence was reconstructed using scales from equations (6.42) and (6.43) with the mean reprojection errors 0.28 and 0.29 pxl, respectively. However, the metric upgrade was more successful for the the alternative given by equation (6.43). The reason is perhaps that only one scale is estimated per camera triplet and that is exactly the one inherently consistent with the shape of the reconstruction.

6.5.4 Three-view Correspondences

To demonstrate usage of systems (6.39) and (6.42), we introduce a simple technique for obtaining consistent scaling of image points satisfying overdetermined constraints on projective depths in sense of [113]. Namely we address the situation when more than the minimal set of $m-1$ EGs are used, where m denotes the number of views. A solution to this problem appeared in [65] (section 6.3.6). It was based on estimation of scales α^{ij} attached to each ij-EG. However, α^{ij} were estimated jointly with all consistent projective depths which are many: in order of hundreds of thousands for hundreds of views. Thus, a large system of equations had to be solved.

Here we study a specific situation in which the α scales (section 6.3.6) do not have to be estimated at all, because they can be set arbitrarily. The reason is that the consistent scaling can be achieved afterwards using the techniques from the previous section. Consider all image points are observed in the same images. For the purpose of explanation, consider images 1-3. When considering a low number of views, the $\mathtt{A}$ matrix from system (6.21), *p*63, is small and $\mathtt{B} = (\mathtt{A}^\top\mathtt{A})^{-1}\mathtt{A}^\top$ can be computed just once for all points. Then, the consistent depths are found as

$$\lambda_p^i = \exp(\overline{\lambda}_p^i)$$

where

$$\begin{bmatrix} \overline{\lambda}_1^1 & \dots & \overline{\lambda}_n^1 \\ \overline{\lambda}_1^2 & \dots & \overline{\lambda}_n^2 \\ \overline{\lambda}_1^3 & \dots & \overline{\lambda}_n^3 \end{bmatrix} = \mathtt{B}\mathbf{b}$$

where the p^{th} column in $\mathbf{b}$ contains logarithm of the ratios of the depths estimated in each EG:

$$\begin{bmatrix} \log \frac{\gamma_p^{12,1}}{\gamma_p^{12,2}} \\ \log \frac{\gamma_p^{23,2}}{\gamma_p^{23,3}} \\ \log \frac{\gamma_p^{13,1}}{\gamma_p^{13,3}} \end{bmatrix}.$$

As mentioned above, α's can be set arbitrarily. The system is overdetermined, thus different α's cause only change in consistent row scaling of the depths between images. Nevertheless, this is no problem, because before factorization (see section 6.3.5), the measurement matrix is rebalanced anyway [113].

The Beryl scene was reconstructed using camera scales from (6.42). It was observed that the repaired scales lead to a more successful metric upgrade [78] meaning that smaller errors in intrinsic camera parameters were obtained. The focal length got closer to the true one. The mean reprojection error increased by 1%. However, it is no lost compared to making the metric upgrade easier. Better results may be caused by better handling of noise.

A possible application of the presented technique is a multiview reconstruction (i) from many high precision EGs from dense stereo [12, 52] or (ii) from EGs from tracking (Pascal Fua [19]). It was not tried as better methods were developed (from section 6.7).

6.6 Merging Panoramas, or "A Successful Approach for the ICCV'05 Contest"

This section presents a technique for estimating focal length and a homography aligning two images in a panorama and a method for aligning all images in a panorama given pairwise alignments. Triplets of panoramas are reconstructed and merged into the 3D reconstruction of the whole scene. A technique for 3D reconstruction of view triplets by making camera rotations consistent is presented. Camera translations are estimated afterwards so that points get in front of cameras. At time of the contest, Kahl's method [44] was not known, which solves the translation and scale estimation problem very well. It was used in all the following methods/sections. The triplet with the highest support is selected. From the remaining triplets with a two-view overlap with the actual reconstruction, the strongest triplet is selected and merged with the actual reconstruction by making rotations and afterwards translations consistent. This is repeated until all panoramas are reconstructed. One-view overlapping triplets are used if necessary. It is demonstrated on difficult image sets that high precision of the reconstruction is reached.

ICCV'05 Vision Contest

In the ICCV'05 Vision Contest [4], a set of photographs was given some of which were given also GPS position. The task was to estimate GPS positions of the remaining images as precisely as possible. Several test sets were available. In the qualifying round, a new data set should be reconstructed within the time limit of a few days. The best five teams took part in contest finals with the time limit of one day.

Our Method

It turned out that for such sparse correspondences (due to small image overlap and low texture) as those in the contest (see figure 6.15) it is necessary to use knowledge of internal camera calibration already in the very early stage of reconstruction for correspondence verification. Till that time, camera internals were used only at the very end in the projective-to-metric upgrade (section 6.3). The six-point algorithm [111] with freely available code was used in RANSAC to estimate the focal length from all available overlapping image pairs (see section 5.3). The implementation of the five-point algorithm [77] was based on the code for the six-point algorithm. Both algorithms provided pair-wise metric reconstructions, which were glued in a similar way to [65] (section 6.3) but in the metric 3D-space instead of the projective 4D-space. Thanks to this the complicated and instable search for the plane at infinity is not needed [33].

Many image pairs in the contest sets formed a panorama. Such image pairs were detected as homographies [16] and explained as camera rotations. These pairs were glued into panoramas by estimating consistent rotations in a similar way to [65] (section 6.3). Thanks to detection and building panoramas we reached entirely precise localization of all images which were a part of some panorama with known GPS position.

The problem of gluing three pairwise reconstructions without three-view correspondences was solved. Such situation may happen when images have a small overlap. If some two (of three) cameras are very close to each other, their mutual position is badly conditioned (relatively to the third camera). It is not known if camera one is on the left- or right-hand side of camera two. Therefore, both configurations are tried and the one with larger support after the non-linear optimization (BA) is used. This is related to the problem when bundle adjustment converges to a false local minimum due to depth reversal. As suggested by [114], this can be avoided by reflecting the depth of the first model solution about the xy-plane, restarting the bundle adjustment, and selecting the solution with the best final reprojection error.

If such complicated situations are many, it is unlikely that mutual camera orientations are correctly "guessed" in all pair- and triplet-wise reconstructions, which would be needed for

Figure 6.15: Images 14 and 17 in the ICCV'05 Contest Test Set 2. Small image overlap and low and unstable texture (reflections in windows) make the correspondence problem challenging.

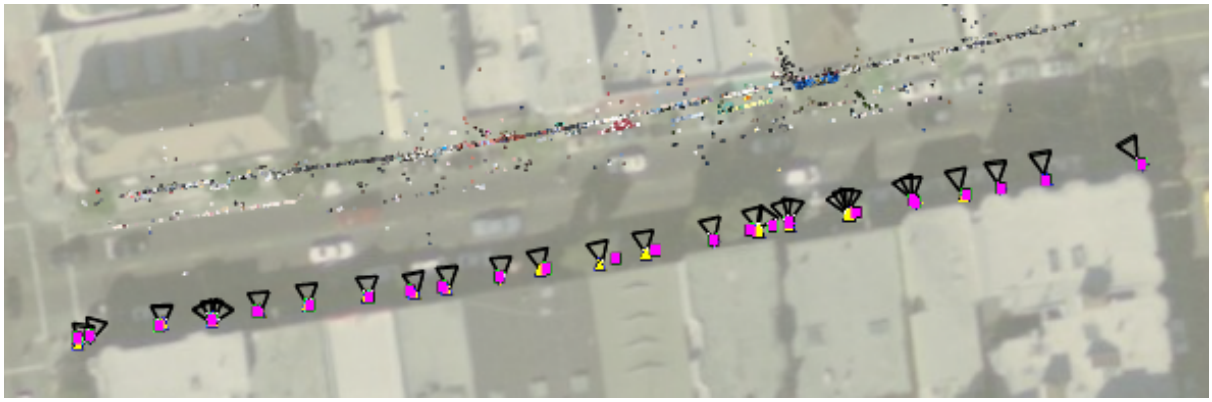


Figure 6.16: The result of our (CMP) team in the qualifying round of the contest. All answers had less than 2 meters in error, in fact, our RMS error was even lower (probably closer to 0.5m although it was not computed by the organizer). Courtesy Richard Szeliski, Microsoft Research.

obtaining a correct gluing by the way "all at once" [65]. Therefore, partial reconstructions are glued gradually while mutual camera position can be switched during adding the next triplet within both the actual camera set and the camera triplet being added.

Our (CMP) team ranked second in the contest finals [59]. We won the qualifying round, see figure 6.16 and the discussion at the end of this section. The complicated switching of camera positions was not needed anymore as a method for translation estimation using SOCP (second order cone programming) appeared just after the contest [44]. It was exploited in all later methods (sections from 6.7, see figure 6.20).

Using Panoramas

Advantages of using panoramas are (i) higher accuracy of the 3D reconstruction if panoramas are correctly recognized (ii) simplification of matching (by merging tentative matches between image pairs one gets much more tentative matches between the panoramas built from the images) and (iii) the 3D reconstruction problem as smaller problem is solved (some cameras merged into panoramas). On the other hand, distinguishing if an image pair is related by a

camera rotation or a full 3D camera movement is a difficult problem which can harm the whole process if a wrong answer is given. Therefore, refinement by bundle adjustment should be used during aligning images in a panorama.[30]

Panoramas can be used in two representations: either as a perspective image (planar sheet in some distance from the camera center) or as vectors on a unit sphere. The latter has the advantage of capturing more than 180 degrees including full omni-cameras.

We used the following simple technique for detection of panoramas. We have not used model selection [121, 120, 48, 94, 90] as we had no code available. EGs with at least some minimal support (30 inliers) are decided if the images are related by a camera rotation or a full 3D camera movement in the following way. If a dominant plane is detected using [16] and the homography support, $N_{\mathtt{H}}$, is at least ϵ ($\epsilon = 90\%$) fraction of the EG support, $N_{\mathtt{F}}$, that is $N_{\mathtt{H}} \geq \epsilon N_{\mathtt{F}}$, the hypothesis on camera rotation is tested. It is done so that both focal length and camera rotation are estimated linearly, as will be explained lower in section 6.6.1. After that, support of the new model is counted. If most of the points have reprojection error below some threshold, the two images are assigned the same panorama.

Then, images are grouped into groups according to their pairwise assignments. There may appear inconsistencies among images such that images i and j should be a part of a panorama because camera rotations ik and jk but pair ij is assigned a full 3D model. The inconsistencies are solved by iterative removal of the pairs which can be least expected to be correct, i.e. with high errors and low support.

6.6.1 Aligning Two Images in a Panorama

Suppose a homography relating two images is given by, e.g., RANSAC [16]. If the two images are a part of a panorama, then the homography, $\mathtt{H}$, can be expressed using camera rotation, $\mathtt{R} \in \mathbb{R}^{3\times3}$, as

$$\mathtt{H} \quad = \quad \mathtt{K}\mathtt{R}\mathtt{K}'^{-1}\beta \tag{6.44}$$

where β is the scale of the homography and $\mathtt{K}$ and $\mathtt{K}'$ are the internal calibrations of the two cameras (same as equation (4.4), *p*20):

$$\mathtt{K} \quad = \quad \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \mathtt{K}' \quad = \quad \begin{bmatrix} f' & 0 & 0 \\ 0 & f' & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.45}$$

where f and f' are the unknown focal lengths.[31]

Equation (6.44) can be rearranged to

$$\mathtt{R} \quad = \quad \mathtt{K}^{-1}\mathtt{H}\mathtt{K}'\frac{1}{\beta}. \tag{6.46}$$

The orthonormality constraint of the $\mathtt{R}$ rotation,

$$\mathtt{R}^\top\mathtt{R} \quad = \quad \mathtt{I},$$

where $\mathtt{I}$ is the 3×3 identity matrix, can be rewritten using (6.46) as

$$\mathtt{K}'\mathtt{H}^\top\mathtt{K}^{-1}\mathtt{K}^{-1}\mathtt{H}\mathtt{K}' \quad = \quad \mathtt{I}\beta^2. \tag{6.47}$$

[30] this was not implemented

[31] Note that image coordinates can be always transformed so that the transformed cameras satisfy (6.45) once the camera intrinsics are known up to the focal length.

Equation (6.47) can be simplified to

$$\mathtt{H}^\top \mathtt{K}^{-2} \mathtt{H} \;=\; \mathtt{K}'^{-2} \beta^2,$$

which can be rewritten as

$$\mathtt{H}^\top \begin{bmatrix} p & 0 & 0 \\ 0 & p & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathtt{H} \;=\; \begin{bmatrix} q\alpha & 0 & 0 \\ 0 & q\alpha & 0 \\ 0 & 0 & \alpha \end{bmatrix} \tag{6.48}$$

where $p = f^{-2}$, $q = f'^{-2}$, and $\alpha = \beta^2$.

System of equations (6.48) consists of 3×3 equations. One can estimate α from equation $(3,3)$. Then, knowing α, one can estimate p and q from system of equations $(1,1)$ and $(2,2)$.

Another possibility, which was also used in the contest, is to first estimate p from the overdetermined system of equations $(2,1), (3,1)$ and $(3,2)$. Assuming the same focal lengths, one can set $q = p$ and estimate α from the overdetermined system of equations on the diagonal, i.e., $(1,1), (2,2), (3,3)$. Knowing α, p can be reestimated using all the nine equations achieving so higher consistency with all the homography parameters. Note that by using the overdetermined systems one can better handle errors in the estimated homography $\mathtt{H}$ caused by image noise and radial distortion.

Aligning Multiple Images in a Panorama

Given pairwise alignments (rotations) of some image pairs in a panorama, it is possible to align all images in the panorama (rotation registration). The method was later published in [66], see equations (6.52), *p*89 in section 6.7.2.

Discussion

There are multiple reasons why our method did not win the contest:

- Only the strongest view triplet was used to connect the actual reconstruction with the image being added. Instead, all triplets connecting the actual reconstruction with the added image should be used.
- Only a few steps in bundle adjustment could be done due to slow MATLAB implementation. Ten or hundred times more steps can be done using an implementation in C, such as [54].
- The camera focal length has not been calibrated using the calibration grids available at the contest web page. No radial distortion removal has been applied.

6.7 Metric Gluing, or "How to Achieve a Good Reconstruction from Bad Images"

This section[32] presents a technique for estimating a multi-view reconstruction given pair-wise metric reconstructions up to rotations, translations and scales. The partial reconstructions are glued by the following three step procedure: (i) Camera rotations consistent with all reconstructions are estimated linearly. (ii) All the pair-wise reconstructions are modified according to the new rotations and refined by bundle adjustment while keeping the corresponding rotations same. (iii) The refined rotations are used to estimate camera translations and 3D points using Second Order Cone Programming by minimizing the L_∞-norm [44]. We present a criterion for evaluating importance of an epipolar geometry in the context of the overall 3D geometry. The estimated importance is used to reweight data in the reconstruction algorithm to better handle unequiponderantly captured images. The performance of the presented method is demonstrated on difficult wide baseline image sets.

A step towards automatic reconstruction procedure providing a high quality reconstruction from a difficult image set is made. This task is difficult and has been extensively studied for last two decades (chapter 3). In this section we particularly focus on the following problems:

- An extreme occurrence of the missing data. In practice it may happen that there are no points visible in more than two images in a (sub)set of images, see figure 6.14, *p*75.
- Degenerate situations like large planes in the image may prevent RANSAC [31] from choosing the non-degenerate full 3D model but with smaller support than of the model given by homography [16]. Another degenerate situation arises when images are taken from one place just by zooming or rotating the camera.
- Some parts of the object may be captured on many more photographs than some other parts. We have observed that in such a case, without a good estimate of the focal length, the standard bundle adjustment [31] may break the reconstruction into discontinuous parts, see figures 6.17 and 6.19.

Note that one might argue that in order to obtain a good model, one should take appropriate images rather than accept bad ones, and that even a non-expert can be easily guided to take better images. Nevertheless, cases where bad images have to be accepted may indeed be relevant sometimes, e.g. when modeling a building or monument that can not be easily photographed from all around.

This section presents an algorithm with the advantage that no point visible in three or more views is required. It consists of three steps: (i) a linear estimate of consistent rotations (ii) refinement of the estimate and (iii) camera translation and point recovery using SOCP [44]. Its first step is a variation on [65] for the metric case. In this section, the partial reconstructions are glued via cameras. All partial reconstructions are glued at the same time thus exploiting all data equiponderantly.

Our method differs from [65] (section 6.3) in that the estimated transformation between the coordinate system of a partial reconstruction and the coordinate system of the reconstruction of all cameras is more simple: only rotation, translation and scale are needed instead of full projective 4×4 homographies in [65][33]. Second advantage of metric over projective reconstruction is that no metric upgrade [78, 33] is needed. Metric upgrade becomes a very difficult task

[32] Most of the section was published in [66]. Richard Szeliski from Microsoft Research provided the ICCV'05 Contest data. Jana Kostková from the Czech Technical University provided routines for dense stereo. Our bundle adjustment routine was based on publicly available software [54].

[33] Method [65] (section 6.3) must fail on the Head data because there is not enough relations between cameras 8–10 (see figure 6.19) which would determine all the fifteen parameters of the projective transformation. Indeed it failed: the 3D model (after the metric upgrade) was split into two parts.

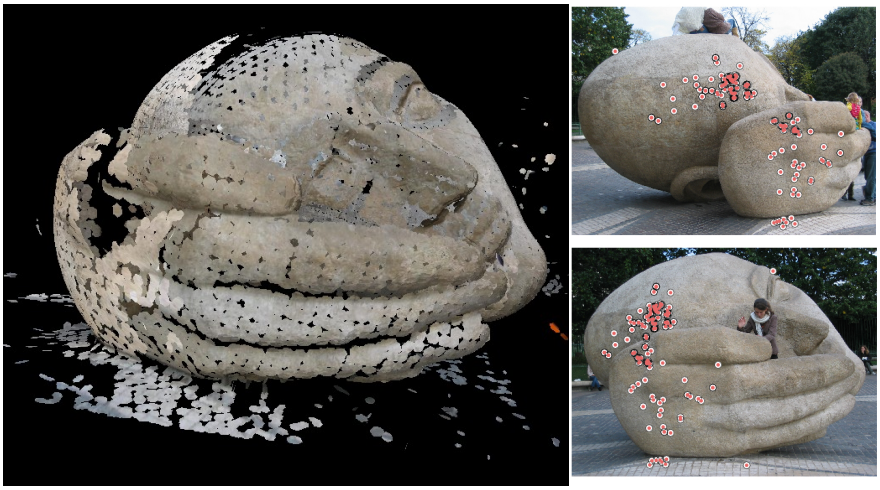


Figure 6.17: Incorrect reconstruction provided by the standard bundle adjustment [31] for non-equiponderantly captured images of the Head set (left) low amount of correspondences on the back of the hand. Only inliers w.r.t. the final 3D-model shown (right).

on data like presented here. Even if the projective reconstruction is successfully transformed so that a reasonable subset of points gets in front of cameras, the internal camera parameters get rather far from what is desired (e.g. square pixel) making bundle adjustment prone to stucking in local minima. In [65], the algebraic (SVD) error is minimized instead of the reprojection error. Compared to this method, minimizing inconsistency between rotations in the metric space promises reaching a higher stability.

It is known that rotation can be estimated first and translation can be estimated using it afterwards [128]. In [128], differences between rotations parameterized using quaternions were non-linearly minimized while using some additional constraints like vanishing points. In our method, 3×3 matrices are used to parameterize rotations. Although these matrices describe the class of all homographies, the rotation obtained as the closest rotation to such homography in the least squares gives results sufficient for our task, as rotations are subsequently refined in step (ii).

The method on linear estimation of consistent camera rotations will be explained in section 6.7.2 and rotation refinement in section 6.7.3. Obtaining consistent camera translations and scales comes in section 6.7.4. A criterion on influence of an EG on the overall shape will be described in section 6.7.5 and its application will be shown in section 6.7.5. Experiments are reported in section 6.7.6.

Problem Formulation

Suppose m images captured using a standard camera with focal lengths known[34] up to an unknown overall scale factor. Points of interest are found in all images and matched between all image pairs using a similarity measure (see more details on our experiments in section 6.7.6) There are mismatches in image measurements. The goal is to recover cameras and 3D points.

[34] e.g., from the EXIF header of the JPEG file

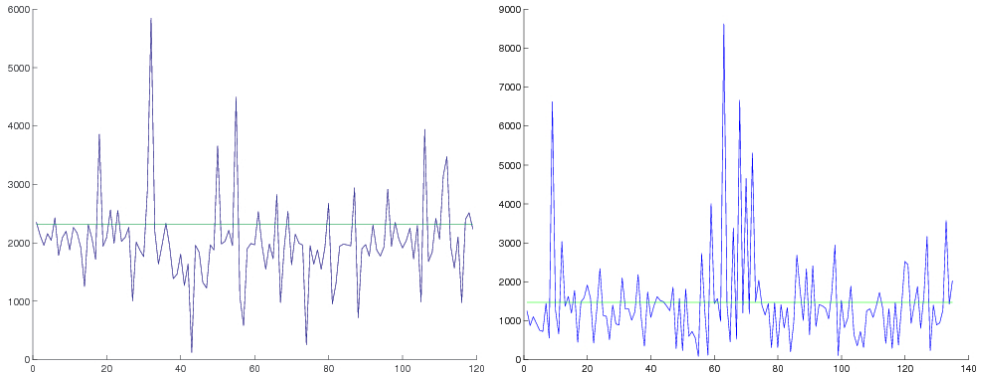


Figure 6.18: Focal length estimates from the six-point algorithm used as described in section 6.7.1 on the Head and Contest sets. The horizontal line corresponds to the estimate of the overall focal length.

The Method

The six-point RANSAC [111] is applied to all image pairs. When the two corresponding focal lengths differ, one of the images is rescaled so that the focal lengths become the same. The overall scale of the focal length is then estimated as the mean of the estimates given by the six-point algorithm weighted by the square of the EG support. Then, the five-point algorithm [77] is run on all image pairs.

6.7.1 RANSAC on EG and a Dominant Plane

In this paragraph, simple methods from [66] are explained. More sophisticated approaches are described in sections 5.3 (focal length estimation in log-scale) and 4.3 (robust RANSAC). An epipolar geometry unaffected by a dominant plane is found using [16]. The inliers are used as the pool for drawing samples in calibrated RANSACs. This scheme is applied to the six-point algorithm [111] as well as to the five-point algorithm [77]. Due to unstability of estimate of the focal length [111], the degenerate samples (all points in the dominant plane provided by [16][35]) should be detected and thrown away. If all points lie in a plane or two images form a panorama, the correct focal length cannot be estimated. Thanks to the small amount of outliers in the pool, the five-point RANSAC has a bigger chance to find the correct non-degenerate EG, especially with a substantial error in the focal length. It can be seen in figure 6.18 that estimates of the overall scale of the focal length is quite unreliable (see also beginning of section 6.7.5).

6.7.2 Consistent Rotations

Let $\mathtt{A}^{\mathbf{i}}$ denote the submatrix of $\mathtt{A}$ composed of elements in rows $\mathbf{i}$. Omitting superscript means taking all rows. Suppose T pair-wise metric reconstructions are given for camera indices $\mathbf{i}_1$, $\mathbf{i}_2$, ..., $\mathbf{i}_T$ where $\mathbf{i}_t \in \{1, 2, ..., m\}$, $|\mathbf{i}_t| = 2$ for each t.[36] Let the cameras of the t^{th} reconstruction be denoted as $\tilde{\mathtt{P}}_t$, $\tilde{\mathtt{P}}_t \in \mathbb{R}^{6\times 4}$. Each reconstruction is generally in a different coordinate system. It has been shown in [65, equation (4)] that the coordinate systems are related by homographies, $\mathtt{H}_t$, which can be linearly estimated together with a set of all cameras, $\mathtt{P} \in \mathbb{R}^{3m\times 4}$, in the same

[35] One might check if all the points lie on another (smaller) plane. Note that such samples are unlikely to win in RANSAC.

[36] In this section, only equations for pair-wise reconstructions are shown. Equations for triplets, etc. are similar.

(global) coordinate system:

$$\begin{aligned} \tilde{\mathtt{P}}_1\mathtt{H}_1 &= \mathtt{P}^{\mathbf{i}_1} \\ &\vdots \\ \tilde{\mathtt{P}}_T\mathtt{H}_T &= \mathtt{P}^{\mathbf{i}_T}. \end{aligned} \tag{6.49}$$

For cameras are calibrated, after denoting indices of the t^{th} pair-wise reconstruction as i and j, $\{i, j\} = \mathbf{i}_t$, the t^{th} equation in (6.49) can be written as

$$\begin{bmatrix} \mathtt{K}^i & [\ \mathtt{I} & | & \mathtt{0}\ ] \\ \mathtt{K}^j & [\ \mathtt{R}_t & | & \mathbf{t}_t\ ] \end{bmatrix} \begin{bmatrix} \mathtt{h}_t & \mathbf{u}_t \\ \mathtt{0}_{1\times 3} & s_t \end{bmatrix} = \begin{bmatrix} \mathtt{K}^i & [\ \mathtt{R}^i & | & \mathbf{t}^i\ ] \\ \mathtt{K}^j & [\ \mathtt{R}^j & | & \mathbf{t}^j\ ] \end{bmatrix}. \tag{6.50}$$

Here, the matrix of internal parameters, $\mathtt{K}^i \in \mathbb{R}^{3\times 3}$, is known with focal length estimated as described in section 6.7.1, $\mathtt{R}$ is a 3×3 rotation, $\mathbf{t} \in \mathbb{R}^3$ is a translation, and $\mathtt{I} \in \mathbb{R}^{3\times 3}$ is an identity matrix. Homographies $\mathtt{H}_t$ simplified to rotations $\mathtt{h}_t$, translations $\mathbf{u}_t$ and scales s_t. To simplify notation, rotation and translation of the first camera in each partial reconstruction has been transformed to identity and zeros, respectively, by applying appropriate rotation and translation beforehand.

After multiplying each triplet of rows by the corresponding $\mathtt{K}^{i^{-1}}$ from the left, system (6.50) becomes

$$\left[\begin{array}{c|c} \mathtt{I} & \mathtt{0} \\ \mathtt{R}_t & \mathbf{t}_t \end{array}\right] \begin{bmatrix} \mathtt{h}_t & \mathbf{u}_t \\ \mathtt{0}_{1\times 3} & s_t \end{bmatrix} = \left[\begin{array}{c|c} \mathtt{R}^i & \mathbf{t}^i \\ \mathtt{R}^j & \mathbf{t}^j \end{array}\right]. \tag{6.51}$$

By writing only the first three columns of equation (6.51), one obtains:

$$\begin{aligned} \begin{bmatrix} \mathtt{I} \\ \mathtt{R}_1 \end{bmatrix} \mathtt{h}_1 &= \begin{bmatrix} \mathtt{R}^i \\ \mathtt{R}^j \end{bmatrix} \\ &\vdots \\ \begin{bmatrix} \mathtt{I} \\ \mathtt{R}_T \end{bmatrix} \mathtt{h}_T &= \begin{bmatrix} \mathtt{R}^i \\ \mathtt{R}^j \end{bmatrix} \end{aligned} \tag{6.52}$$

as translation $\mathbf{t}_t$ is multiplied with zeros in the middle matrix in equation (6.51). One could try to simplify system (6.52) by fixing the overall coordinate system by setting $\mathtt{h}_1 = \mathtt{I}$. Then, using the first equation in (6.52), one could set $\mathtt{R}^i = \mathtt{I}$ and $\mathtt{R}^j = \mathtt{R}_t$, etc. However, in case of errors in image measurements, no precise solution to (6.52) exists. Therefore, this way would lead to unequiponderant spreading of the error between partial reconstructions as some equations are fulfilled and some are not.

A solution to system (6.52) in the least squares was published in [65, section 2.1] (section 6.3.1). See it for details on the solution using MATLAB's EIGS and its numerical behaviour. It provides the closest solution to all partial rotations in the least squares. The found solution does not represent rotations (the 3×3 matrices are not orthonormal), but as it is close to the true rotations of partial reconstructions, it is close to some true rotations as well. The true rotations, $\bar{\mathtt{R}}^i$, can be found as the closest rotation in the least squares, e.g., as $\bar{\mathtt{R}}^i = \mathtt{U}\mathtt{V}^\top$ where $\mathtt{R}^i = \mathtt{U}\,\mathrm{diag}(\sigma_1, \sigma_2, \sigma_3)\mathtt{V}^\top$ is the SVD factorization.

The t^{th} equation in (6.52) is weighted by the root of the EG support, as it was done in [65, equation (10)] (equation (6.12)).

6.7.3 Refining Rotations

Rotations in the t^{th} partial reconstruction are replaced by the consistent rotations, $\bar{\mathtt{R}}^{\mathbf{i}_t}$, and translations are modified accordingly using $\mathtt{h}_t$[37]. Due to errors in image measurements, the

[37] Better results were achieved without projecting $\mathtt{h}_t$ onto the space of rotations.

rotation between consistent rotations $\bar{\mathtt{R}}^i$ and $\bar{\mathtt{R}}^j$, $\{i,j\} = \mathtt{i}_t$ for some t, does not equal the rotation between the two cameras in the t^{th} original partial reconstruction, $\bar{\mathtt{R}}^{i\top}\bar{\mathtt{R}}^j \neq \mathtt{R}_t$. As a consequence, reprojection errors grow after making rotations consistent. Hence, refinement using bundle adjustment is desired. Before that, it is necessary to either rotate points in the t^{th} partial reconstruction by $\mathtt{h}_t^{-t}$ or triangulate them using the new rotations. The latter is preferred, as lower reprojection errors are achieved.

Then, the triangulation using the Sampson's approximation [31] is computed. For points reconstructed behind at least one camera [132], the triangulation is computed again using SOCP in the L_∞-norm [44]. The reason for doing the Sampson's approximation first is that it is much faster. Then, BA was applied on the t^{th} partial reconstruction while keeping rotations constant. When some of the points got behind a camera during the BA, both camera translations and points were estimated using SOCP [44]. SOCP was used here only as an emergency solution as it is time consuming compared to BA.

Finally, full bundle adjustment on all partial reconstructions at the same time with corresponding rotation parameters kept the same is run. The overall scale of focal lengths is varied. Such a bundle adjustment (BA) is something in between T independent BAs of T partial reconstructions, each with two cameras, on one hand and standard BA with all m cameras in the same coordinate system on the other hand. Here, $2T$ cameras share only rotations while translations are still inconsistent. This way provides a higher flexibility allowing to change translation in the t^{th} partial reconstruction almost independently of translations in the remaining reconstructions. More precisely, translations influence themselves via shared rotations, which is however a more free connection than demanding consistency between translations. This approach can thus be expected to be less prone to stucking in a local minimum than the standard BA.

Some EGs with small support may be found even on image pairs with no overlap. These EGs are easily detected after several (20) steps of BA as those with no inliers with respect to the desired accuracy (1 pixel).

6.7.4 Consistent Translations and Scale

A straightforward way for obtaining global scales and translations is to estimate them together with rescaling and translation of each partial reconstruction in a similar way as in section 6.7.2. However, our results when using this approach were not satisfactory. The reason why this approach worked well for rotations is perhaps that there are no significant differences in magnitudes of the variables (there are just orthonormal 3×3 matrices in equation (6.52)). On the other hand, translations can have large differences in magnitudes across partial reconstructions.

Therefore, the state-of-the-art SOCP method [44] was used to estimate both camera translations and points. It gives good results as the reprojection error is minimized while keeping all points in front of cameras. The fact that the L_∞-norm is minimized is not a problem as final bundle adjustment minimizing the L_2-norm is run anyway.

Handling Mismatches

To add robustness to mismatches, only some points (from more than 10^5 in our experiments) are sampled for bundle adjustment and SOCP in section 6.7.4. To capture the overall geometry, points are sampled so that from each image pair having nonzero points in common, 90% of points with lowest reprojection errors are chosen. By this simple way most mismatches are removed while removing only a reasonable amount of inliers and capturing the overall geometry without a complicated threshold estimation.

6.7.5 Handling Unequiponderant Data

In our setup, the focal length for camera calibration was estimated from the data not accurately (there was about 5% error on the Head scene). If images of the scene were captured equiponderantly so that all sides of the object occupied roughly the same amount of the image data, the standard bundle adjustment converged to a satisfactory minimum. However, this is not the case of the data used here. All parts of the head sculpture were captured on quite many images except the back of the hand where only six images (11-13, 20-22, see figure 6.19) were taken with wide baseline causing fewer matches. The error in the estimate of the focal length (and perhaps also the radial distortion) caused in the standard bundle adjustment that the well covered data overweighted the small contribution from the back of the hand. As a result, the hand and the head in place above it are split into two discontinuous surfaces, see figure 6.17.

Our approach to avoid such failures is to find which EGs are more important for the geometry of the overall 3D model and to weight such EGs more. Distinguishing which EGs are more important for 3D is hard even if some rough 3D model is given because there may be various degeneracies like dominant planes and camera rotations. To do it thoroughly, one should consider that on one hand wide baseline pairs provide better conditioned 3D estimates but on the other hand they have smaller support due to large camera movement.

Two terms should be distinguished: *importance* of an EG and *quality / reliability* of an EG. An EG will be called *weak* if it is not reliable. Even a weak EG can be important, thus if it was removed, the overall geometry would change much. On the other hand, a strong EG does not have to be necessarily important, consider for instance two same images. Here by an EG it is meant relative position and orientation of the camera pair, which is equivalent to the epipolar constraint plus camera calibration.

The larger the support is, the more reliable the EG could seem to be. However, if all correspondences lie in a small area in an image, the constraint on the overall geometry provided by the EG is rather weak. Quality / reliability of an EG can be evaluated by perturbation of EG parameters [22, 57].

One way of determining the importance of an EG could be to reconstruct / refine the 3D reconstruction without the EG and observe how much the reconstruction changes. If it changes much, the EG is important for the overall geometry. This would be repeated for all EGs. Note that (i) this approach is very computationally demanding and (ii) it cannot not work in presence of wrong EGs as they (as well as the really important EGs) would result in large changes once removed. See section 6.8 for detection of non-existent EGs.

Our method is rather simple but worked well on our data. It is based on finding shortest (and slightly longer) paths in a graph induced by known EGs. Each vertex of the graph stands for a camera. Two vertices are connected by an edge iff there is a known EG between the corresponding cameras. We have observed that

Principle 1 *For estimating relative positions of any two cameras, the most important EGs are those which lie on the shortest paths between the two cameras.*

Shorter paths seem to be more important than longer ones because noise in each additional camera along the path increases uncertainty in the 3D geometry between the two cameras.

Let the *graph of known EGs*, $G = (V, \mathtt{E})$, be defined as a set of vertices, V, $V = \{1 \dots m\}$, and an adjacency matrix, $\mathtt{E} \in \mathbb{R}^{m \times m}$, where V corresponds to cameras and $\mathtt{E}(i, j) = 1$ when an EG is defined between cameras i and j, otherwise $\mathtt{E}(i, j) = 0$. In our current implementation, an EG is defined if it has at least some minimal support (30 inliers).

The task is to estimate importance of all EGs. It will be stored in *EG importance* matrix $\mathtt{S} \in \mathbb{R}^{m \times m}$. Between each pair of vertices, all shortest (and slightly longer) paths will be found in a breadth-first-search manner as will be explained below. All such paths contribute to the

importance of EGs (associated with the edges) through which they pass. All contributions are summed up in the EG importance $\mathtt{S}$.

It is not sufficient to find just one shortest path between two vertices in the graph. The reason is that if more shortest paths exist, all participate in constraining the 3D geometry between the two cameras. Thus, Floyd-Warshall's algorithm [103] is not usable as it finds just one shortest path between two vertices, although it has low complexity $O(m^3)$.

Finding All Shortest Paths

A *path* means here a sequence of adjacent vertices and edges where both can appear multiple times. In a *simple path*, all vertices and edges are distinct.

It is well known in graph theory that $\mathtt{E}^k(i,j)$ equals the number of all paths of length k between i and j where $\mathtt{E}^k$ is the k^{th} power of $\mathtt{E}$. On a complete graph, i.e. $\mathtt{E}(i,j) = 1$ iff $i \neq j$, it can be easily shown that $\mathtt{E}^k(i,j) \geq (m-2)^{k-1}$ for $i \neq j$. Due to the exponential growth of the number of paths with their length, finding all paths between two vertices followed by adding some weight to the $\mathtt{S}$ matrix on edges along the shortest paths is infeasible.

Our strategy is not to track all paths one by one (they are too many) but to track all paths of length k from vertex f to the remaining vertices at the same time. For each vertex, t, all paths of length k from f to t are registered. As the only desired output is the $\mathtt{S}$ matrix, i.e. some weights on graph edges, it is sufficient to register not all particular paths but only the number of paths leading via each edge. At each vertex, $t \in \{1 \dots m\}$, matrix $\mathtt{A}_t^k \in \mathbb{R}^{m \times m}$ is stored. Entry $\mathtt{A}_t^k(i,j)$ equals the number of all paths of length k between vertices f and t leading via edge (i,j).

Our algorithm for finding all paths from a given vertex, f, to the remaining vertices works as follows. Paths of length one are registered, i.e. $\mathtt{A}_i^1(f,i) = \mathtt{A}_i^1(i,f) = 1$ for $i \in \text{neighbors}(f)$. At step k, paths of length k are prolonged and stored in matrices $\mathtt{A}_i^{k+1}$. The shortest paths between f and i are in $\mathtt{A}_i^k$ where

$$k^* = \min\{k \mid \mathtt{A}_i^k \text{ is not all zeros}\}. \tag{6.53}$$

Proposition 6 *Matrix $\mathtt{A}_i^{k^*}$ corresponds to shortest paths from f to i for k^* defined in equation (6.53) (which also means they are simple paths).*

Proof. If there was any shorter path of length l, $l < k^*$, matrix $\mathtt{A}_i^l$ would have some non-zero element. Contradiction with the definition of k^*. □

The algorithm is summarized in algorithm 5. Here, norm $|\cdot|$ of a matrix denotes the sum of its elements, $|\mathtt{A}| = \sum_{i,j} \mathtt{A}(i,j)$. The upper bound on complexity of algorithm 5 is $O(m^3 E)$ where E is the number of graph edges when using sparse matrix representation. It is run for all vertices:

initiate matrix $\mathtt{S} \in \mathbb{R}^{m \times m}$ to zeros
for $f \in \{1 \dots m\}$
  $\mathtt{S} = \mathtt{S} + \mathtt{S}_f$ //contribution by paths from f

Thus, the overall complexity is at most $O(m^4 E)$. It results in a fraction of time spent in the reconstruction pipeline.

Note on algorithm 5. The formula for the number, N, of paths of length k leading from f to p, $N = \frac{|\mathtt{A}_p^k|}{2k}$, can be easily found by induction. It also holds $N = \mathtt{E}^k(f,p)$. □

The EG importance, $\mathtt{S}$, found using algorithm 5 on the Head set is shown in figure 6.19a. It turns out that edges close to articulations (here vertices 9 and 22) in the graph gather up most

Input: A graph and a vertex, f. EG reliability matrix, $\mathtt{w}$.
Output: Contribution, $\mathtt{S}_f$, to the EG importance matrix by all shortest and slightly longer paths from f to the remaining vertices. Similarly for contribution, $\mathtt{T}_f$, to the EG reliability-importance matrix.

initiate $\mathtt{A}_i^k, \mathtt{W}_i^k \in \mathbb{R}^{m\times m}$ to zeros for $i,k \in \{1\dots m\}$
for $i \in$ neighbours(f)
  $\mathtt{A}_i^1(f,i) = \mathtt{A}_i^1(i,f) = 1$
  $\mathtt{W}_i^1(f,i) = \mathtt{W}_i^1(i,f) = \mathtt{w}(i,f)$
for $k \in \{1\dots m-2\}$ //prolong paths of length k
  for $p \in \{i \mid \mathtt{A}_i^k$ is not all zeros$\}$
    for $t \in$ neighbours(p)
      $\mathtt{B} = \mathtt{A}_p^k$ //take all paths from f to p
      $\mathtt{V} = \mathtt{W}_p^k$
      $\mathtt{B}(t,p) = \mathtt{B}(p,t) = \mathtt{B}(p,t) + \frac{|\mathtt{A}_p^k|}{2k}$ //prolong to t
      $\mathtt{V}(t,p) = \mathtt{V}(p,t) = \mathtt{V}(p,t) + \frac{|\mathtt{A}_p^k|}{2k}$
      $\mathtt{A}_t^{k+1} = \mathtt{A}_t^{k+1} + \mathtt{B}$
      $\mathtt{W}_t^{k+1} = \mathtt{W}_t^{k+1} + \mathtt{V}\cdot\mathtt{w}(p,t)$
initiate $\mathtt{S}_f$, $\mathtt{T}_f \in \mathbb{R}^{m\times m}$ to zeros
for $t \in \{1\dots m\}\backslash f$
  $l^* = \min\{l \mid \mathtt{A}_t^l$ is not all zeros$\}$ //shortest path to t
  initiate $\mathtt{B}$, $\mathtt{V} \in \mathbb{R}^{m\times m}$ to zeros
  for $k \in \{l^* \dots \lceil \frac{3}{2} l \rceil\}$ //+ slightly longer
    $\mathtt{B} = \mathtt{B} + \mathtt{A}_t^k \frac{2}{|\mathtt{A}_t^k|}$ $\quad$ $\mathtt{V} = \mathtt{V} + \mathtt{W}_t^k \frac{2}{|\mathtt{A}_t^k|}$
  $\mathtt{S}_f = \mathtt{S}_f + \mathtt{B}\frac{2}{|\mathtt{B}|}$ $\quad$ $\mathtt{T}_f = \mathtt{T}_f + \mathtt{V}\frac{2}{|\mathtt{V}|}$

Algorithm 5: Algorithm for finding all shortest and slightly longer paths from a given vertex to the remaining ones.

importance, which is what is desired. However, it also turns out that shortest paths tend to include weak EGs, which are prone to be completely wrong, like EG 2-19 between images with no overlap, see figure 6.19a and 6.19b. Therefore, two extensions are made:

1. So called *EG reliability-importance matrix*, $\mathtt{T}$, is estimated similarly to the EG importance matrix $\mathtt{S}$ by reweighting edges along each path by the corresponding EG reliability. The *EG reliability matrix*, $\mathtt{w} \in \mathbb{R}^{m\times m}$, holds the ij-EG support in $\mathtt{w}(i,j)$. See algorithm 5.

2. Slightly longer (by factor of $\frac{3}{2}$ in our implementation) than the shortest paths are used also.

Both these extensions lead to suppressing of weak EGs which lie along short paths (see figure 6.19c), thus providing higher robustness to mismatches and less sensitivity to the threshold on an acceptable EG.

Using the EG importance

The estimated EG reliability-importance can be used in two ways: (i) First, it can be used to weight the corresponding equation in (6.52) instead of EG support (see end of section 6.7.2) and to weight the data in the BA in section 6.7.3 as well as in the final BA. (ii) Second, the most important EGs can be strengthened by adding appropriate image triplets.

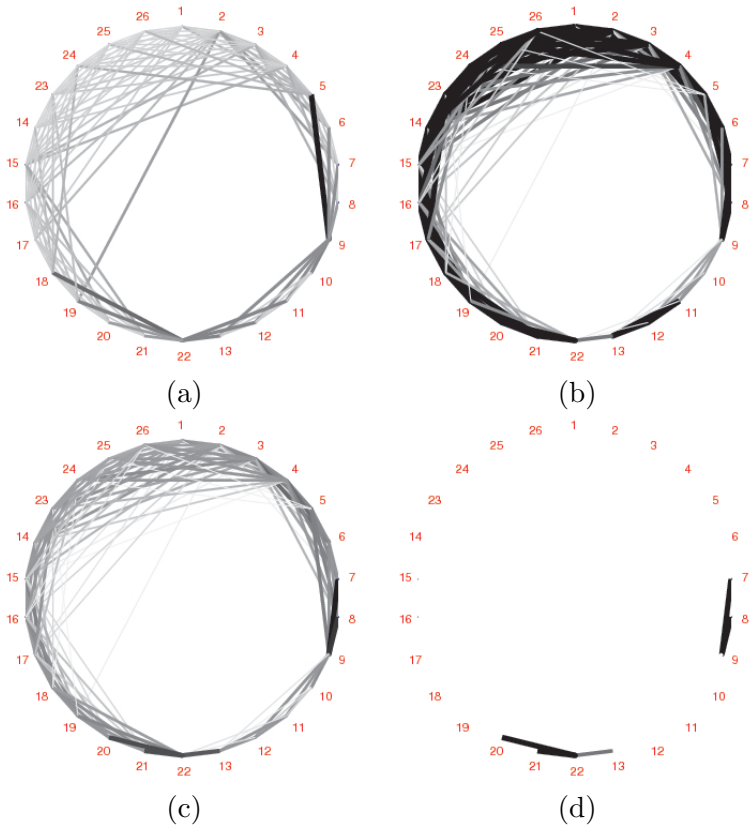
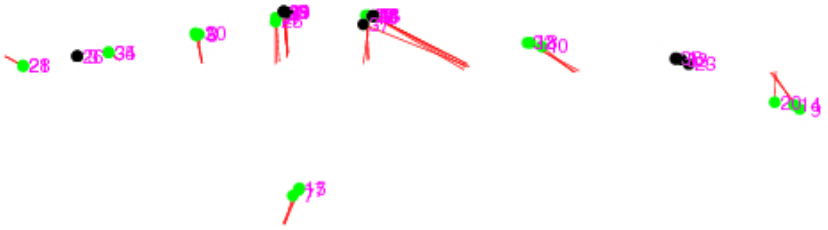


Figure 6.19: Scoring of EGs using all shortest and slightly longer paths on the Head set. (a) EG importance (b) EG reliability (c) EG reliability-importance (d) 4% of the most important EGs. More important/reliable EGs are drawn darker and thicker. The most shortest paths lead via articulations (images 9 and 22). Images are reordered due to visualization.

Unfortunately, important EGs have often small support, thus triplets containing them have even smaller (a point visible in three images must be visible in each image pair). Our solution is to add triplets containing only image i and triplets containing image j. In experiments shown in this section, 4% of the EGs with the highest reliability-importance were chosen (see figure 6.19d). All triplets containing at least one of the images associated with these EGs were taken if all the three EGs were defined. In the Head set, 69 triplets were chosen.

One might use the three-view matches in BA to refine the initial estimate obtained using pair-wise matches as described above. A better way is to exploit the data at the very early stage for obtaining the consistent rotations in section 6.7.2. This is very useful as image triplets provide stronger constraints on 3D geometry. Thus, a better initial estimate of the reconstruction may lead to avoiding some local minima in BA.

Importance of a triplet was estimated as the mean of the reliability-importance of the three associated EGs weighted by the number of the three-view inliers. Partial reconstructions of the chosen image triplets were obtained in the following way. (i) An initial estimate of the camera triplet was estimated from pair-wise reconstructions using the algorithm described above. (ii) A four-point RANSAC was run on three-view matches. For each sample, BA was run on the four points to get the model (three camera matrices). Support was obtained using triangulation. (iii) The reconstruction was refined by BA on two- and three-view inliers.



Avg. Score	< 2 m (5 pts)	< 4 m (4 pts)	< 8 m (3 pts)
4.09	4	16	2

Figure 6.20: Reconstruction of the final round image set from the ICCV05 Vision Contest: (top) top view. The black points correspond to cameras with known GPS positions used to transform the reconstruction into the world coordinate system. Lines join the estimated cameras with the ground truth. (bottom) Score is counted on cameras with unknown GPS positions, see [4].

Figure 6.21: All-around reconstruction of the Head statue (from left): front view using fish-scales, side and overall top views with reconstructed buildings around using point clouds (10% points shown).

6.7.6 Experiments

In experiments reported here, pairwise image matching was done with Local Affine Frames [70] constructed on intensity and saturation MSER regions, LaplaceAffine and HessianAffine [75] interest points.

The Head sculpture was captured on 26 images. A similar image set of 10 images was used in [17] but covering only cca 120 degrees of the circular path around the statue. 91% data is missing in the measurement matrix. Figure 6.21 captures the all-around reconstruction with correct surface and surrounding buildings obtained using the EG reliability-importance from section 6.7.5.

The presented algorithm has been tested on the image set from the final round of the ICCV Computer Vision Contest [4]. This difficult data set contains several panoramas with many camera rotations and dominant planes. Our method achieved mean / maximum reprojection error of 3.01 / 4.87 meters evaluated on the GPS ground truth available at the contest page. Our result with average score 4.09 outperformed the best team in the contest. Cameras' focal length has not been calibrated using calibration grids available at the contest page. No radial distortion removal has been applied in the contest. The results can be seen in figure 6.20. The bending of the reconstruction is caused by imprecise focal length estimation and perhaps also radial distortion. This scene has a linear structure without any cycle around an object like in

the Head set which could enforce strong constraints on focal length. Two most distant cameras with known GPS positions were aligned by a similarity to the ground truth.

To demonstrate quality of the reconstructions, the estimated cameras were used by method [17] to produce dense reconstructions. The results can be seen in figures 6.21 and 6.17.

Besides scenes shown here, our method was tested on other scenes including the Dinosaur sequence with similar results as in [65] (see figure 6.10). See more reconstructed scenes at [2].

Discussion

To handle degenerate situations, one might detect panoramas. However, decision wether two images are related by a camera rotation is difficult especially with an unreliable estimate of focal length.[38] All steps of our algorithm are suited for both degenerated pairs and pairs describing full 3D geometry, which was demonstrated on the ICCVC05 data.

If all camera centers are collinear, the 3D reconstruction obtained using points visible in two images only will not be unique. Then image triplets are needed as well.

Another possible application of the EG importance is detection of most important image pairs for guided matching.

In this section, projective depths from individual EGs were not made consistent as in [65] (section 6.3). The only geometrical features being made consistent were camera rotations and subsequently translations (at the same time with scale). We expect that making depths consistent at the very beginning stage while keeping the reconstructions metric will bring rotations closer to each other and thus enhance the probability of successful reconstruction. This variation is one of the topics of the future research.

Summary

A method for multiview reconstruction based on making rotations consistent using a linear formulation was presented. It can be used for an extreme case of the missing data, i.e. when each point is visible in two images only. The method is capable of dealing with degenerate situations like dominant planes and camera rotation and zooming.

It has been shown that standard bundle adjustment fails on unequiponderantly obtained data with an imprecise estimate of the focal length, but when importance of the data is examined from a global view, correct reconstruction can be obtained. For this purpose, a criterion of importance of an EG on the overall 3D geometry has been formulated using shortest paths in a graph.

6.7.7 Projecting Close to Rotations

This section (unpublished in [66]) presents a method for estimating a transformation which brings the result of system (6.52), *p*89, closer to rotations. Usually it is not needed but sometimes the result of system (6.52) is quite unprecise and such transformation may help.

The simplest way is to use the transformation that aligns the found solution to rotation of, e.g., the first camera in the first partial reconstruction. This solution works when distortions in the found solutions are small. A better way which uses the whole found solution is given lower.

In [119], a transformation making cameras as orthonormal as possible is searched for. It is stated that this problem ([119, equations (16)] equivalent to equation (6.54)), "although non-linear, can be solved efficiently and reliably". In our work, a linear technique is achieved by implying a necessary condition on the desired transformation.

[38] see sections 5.5 and 6.6.1

Let the searched transformation by denoted by $\mathtt{Q}$, $\mathtt{Q} \in \mathbb{R}^{3\times3}$. In is desired that it transforms the approximate rotations, $\mathtt{R}^i$, $i = 1, ..., m$, from system (6.52), *p*89 to rotations, i.e. $\mathtt{R}^i\mathtt{Q}$ should be orthonormal:

$$\mathtt{R}^i\mathtt{Q}\mathtt{Q}^\top{\mathtt{R}^i}^\top = \mathtt{I}_{3\times3} \quad \text{for } i = 1, ..., m. \tag{6.54}$$

System (6.54) is bilinear in unknowns $\mathtt{Q}$. It can be simply linearized by using the observation that matrix

$$\mathtt{Y} := \mathtt{Q}\mathtt{Q}^\top \tag{6.55}$$

is symmetric. System (6.54) can be written as

$$\mathtt{A}\mathbf{y} = 0_{6m\times1} \tag{6.56}$$

where matrix $\mathtt{A}$ has size $6m \times 9$ and vector $\mathbf{y}$ is column-wise vectorization of matrix $\mathtt{Y}$. The symmetricity of matrix

$$\mathtt{Y} = \begin{bmatrix} \mathbf{z}_1 & \mathbf{z}_2 & \mathbf{z}_3 \\ \mathbf{z}_2 & \mathbf{z}_4 & \mathbf{z}_5 \\ \mathbf{z}_3 & \mathbf{z}_5 & \mathbf{z}_6 \end{bmatrix}$$

can be achieved by summing up the columns in matrix $\mathtt{A}$ which correspond to the same values in matrix $\mathtt{Y}$. So, instead of solving system (6.56), the following system is solved:

$$\begin{bmatrix} \mathtt{A}_1 & \mathtt{A}_2 + \mathtt{A}_4 & \mathtt{A}_3 + \mathtt{A}_7 & \mathtt{A}_5 & \mathtt{A}_6 + \mathtt{A}_8 & \mathtt{A}_9 \end{bmatrix} \mathbf{z} = 0_{6m\times1}$$

where $\mathbf{z} = \begin{pmatrix} \mathbf{z}_1 & \mathbf{z}_2 & \mathbf{z}_3 & \mathbf{z}_4 & \mathbf{z}_5 & \mathbf{z}_6 \end{pmatrix}^\top$ encodes the six distinct variables in the $\mathtt{Y}$ matrix.

Transformation $\mathtt{Q}$ can be easily extracted from matrix $\mathtt{Y}$ using, e.g., SVD as $\mathtt{Q} := \mathtt{U}\,\mathrm{diag}(\sqrt{\sigma_1}, \sqrt{\sigma_2}, \sqrt{\sigma_3})$ where $\mathtt{U}\,\mathrm{diag}(\sigma_1, \sigma_2, \sigma_3)\mathtt{V}^\top$ is the SVD factorization of matrix $\mathtt{Y}$. However, correct factorization (6.55) is provided only when $\mathtt{Y}$ is positive definite. It turns out that $\mathtt{Y}$ is often positive definite when a reasonable rotation estimates are given as input, e.g. all datasets in [66] (section 6.7) and [2]. However, on some data with severe mismatches which lead to non-existent EGs, the $\mathtt{Y}$ matrix was not positive definite and thus the rotations could not be recovered. One could try to imply the constraint on positive (semi)definiteness using semidefinite programming, but no such trial has been done yet.

The technique described here is not used in our pipeline currently. The reason is that it is not needed for relative rotation registration (section 6.8.1) which is more stable.

6.8 Robust Rotation and Translation Estimation

It is known[39] that the problem of multiview reconstruction can be solved in two steps: first estimate camera rotations and then translations using them. This section presents robust techniques for both of these steps. (i) Given pair-wise relative rotations, global camera rotations are estimated linearly in least squares. (ii) Camera translations are estimated using a standard technique based on Second Order Cone Programming. Robustness is achieved by using only a subset of points according to a criterion that diminishes the risk of choosing a mismatch. It is shown that only four points chosen in a special way are sufficient to represent a pairwise reconstruction almost equally as all points. This leads to a significant speedup. In image sets with repetitive or similar structures, non-existent epipolar geometries may be found. Due to them, some rotations and consequently translations may be estimated incorrectly. It is shown that iterative removal of pairwise reconstructions with the largest residual and reregistration removes most non-existent epipolar geometries. The performance of the presented method is demonstrated on difficult wide baseline image sets.

A step towards automatic reconstruction procedure from a large number of images is made. This task is difficult and has been extensively studied for the last two decades (chapter 3). In this section, cameras are assumed to be calibrated [111]. In such a setup, pairwise metric reconstructions can be estimated using RANSAC [77] up to similarities. Given these, reconstruction of the whole scene can be obtained by first registering all camera rotations and then translations using them [128, 66]. Mismatches, i.e. wrong point correspondences, cause several problems in such a two-step reconstruction procedure:

1. A *few mismatches* which survived RANSAC cause no difficulty in rotation registration as the relative rotation is only slightly biased. On the other hand, a single mismatch may cause a complete failure of the translation registration when minimizing the maximum reprojection error [44].

2. A *non-existent two-view geometry* may be found when similar or repetitive structures appear on different objects, see figures 6.22 and 6.29. According to our knowledge, no attempt has been done to handle the presence of non-existent pairwise geometries in either rotation or translation registration.

This section presents (i) a method for rotation registration. Two alternatives are presented: using quaternions and using approximate rotations. The latter is simpler and more stable than [66] (section 6.7). (ii) In each pairwise reconstruction, a Gaussian is fitted in the rescaled image space and the most likely mismatches are removed as the most distant points from the Gaussian center. (iii) Only four points carefully chosen among the remaining points are used to represent all points almost equally as all points, thus bringing large speedup and memory savings. (iv) It may happen that the rotation or the translation registration reveals that some EG does not exist. In case rotations were estimated using that EG, they should be reestimated without it as such estimate was biased. It is shown that iterative removal of EGs with the largest residual leads to the removal of most non-existent EGs even for the case of a combination of a least squares and an L_∞ problem. The complete method is summarized in algorithm 2, *p*30.

6.8.1 Rotation Registration

It is supposed that pair-wise metric reconstructions given up to rotations, translations, and scales are provided. A brief description of how they were obtained for the data presented in this section is given in section 6.8.6.

[39] Most of this section was published in [67]. Jan Čech from the Czech Technical University provided routines for dense stereo [12]. Our bundle adjustment routine was based on publicly available software [54].

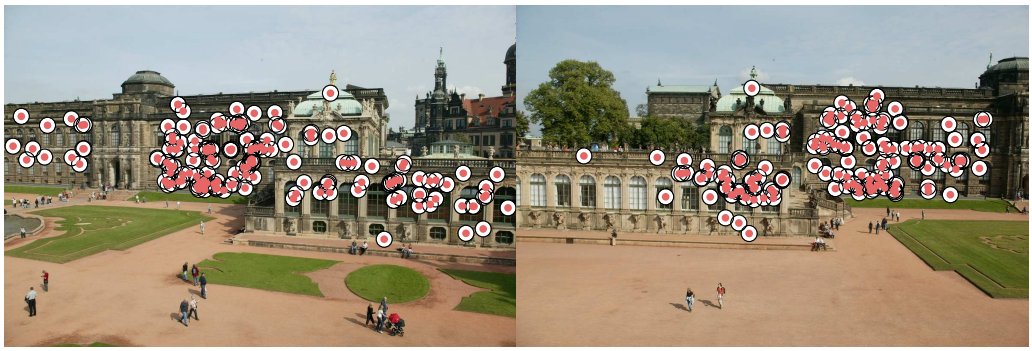


Figure 6.22: A non-existent epipolar geometry (EG) raised by matching similar structures on different buildings in the Zwinger scene. The shown image pair 37-70 has 163 inliers which are 45% of all tentative matches. It would be extremely difficult to find out that this EG does not exist based on the two images only.

The pair-wise reconstruction between views i and j describes the relative rotation between the two cameras, $\mathtt{R}^{ij}$, $\mathtt{R}^{ij} \in \mathbb{R}^{3\times3}$, $\mathtt{R}^{ij}$ orthonormal. The problem of *rotation registration* can be formulated as a search for the registered rotations, $\mathtt{R}^{i}$, $\mathtt{R}^{i} \in \mathbb{R}^{3\times3}$, $\mathtt{R}^{i}$ orthonormal, $i = 1, \ldots, m$, such that relations among them are given by the relative rotations:

$$\mathtt{R}^{j} = \mathtt{R}^{ij}\mathtt{R}^{i} \qquad \text{for all } ij \tag{6.57}$$

$$\mathtt{R}^{i} \text{ orthonormal for } i = 1, \ldots, m \tag{6.58}$$

When $m-1$ relative rotations $\mathtt{R}^{ij}$ are known such that they form a tree graph (with views as vertices connected by an edge whenever the relative rotation between the views is known), system of equations (6.57) is not overdetermined and can be easily solved by fixing the first rotation and chaining the remaining ones.

When at least m relative rotations are given, system (6.57) becomes overdetermined and an exact solution may not exist due to noise in the data. Thus, we solve it in the least squares while satisfying the orthonormality conditions (6.58).

A straightforward solution can be obtained using quaternions. Using them, system (6.57) becomes

$$\dot{r}^{j} = \dot{r}^{ij}\dot{r}^{i} \qquad \text{for all } ij \tag{6.59}$$

where $\dot{r}^{i}$ and $\dot{r}^{j}$ are the unknown quaternions of the i^{th} and j^{th} camera rotation, respectively, and $\dot{r}^{ij}$ is the known relative rotation between cameras i and j. Each quaternion can be thought of as a four-vector, similarly as complex numbers can be thought of as two-vectors. Using known manipulations with quaternions [36], each equation in system (6.59) can be rewritten as

$$\begin{pmatrix} r_0^j \\ r_x^j \\ r_y^j \\ r_z^j \end{pmatrix} - \begin{bmatrix} r_0 & -r_x & -r_y & -r_z \\ r_x & r_0 & -r_z & r_y \\ r_y & r_z & r_0 & -r_x \\ r_z & -r_y & r_x & r_0 \end{bmatrix} \begin{pmatrix} r_0^i \\ r_x^i \\ r_y^i \\ r_z^i \end{pmatrix} = 0_{4\times1} \tag{6.60}$$

where $\dot{r}^{i} = r_0^i + \imath r_x^i + \jmath r_y^i + k r_z^i$ and $\dot{r}^{ij} = r_0 + \imath r_x + \jmath r_y + k r_z$ with $\imath$, $\jmath$ and k as imaginary units. From now on, by the i^{th} quaternion we will mean the four-vector $(r_0^i, r_x^i, r_y^i, r_z^i)^\top$.

There are $4m$ unknowns $\mathbf{r} = (r_0^1, r_x^1, r_y^1, r_z^1, \ldots, r_0^m, r_x^m, r_y^m, r_z^m)^\top$ with constraints (6.60) for each camera pair ij with a known rotation. System of all ij-constraints (6.60) is sparse, thus

it can be solved using, e.g., MATLAB's EIGS. The solution is obtained as a unit vector $\mathbf{r}$. The $\mathbf{r}$ vector is composed from quaternions (four-vectors) $\dot{r}^i$, which are, however, not unit. Fortunately, they can be easily made unit by dividing each by its Euclidean length. This conversion is needed as only a unit quaternion has a corresponding rotation. Then, the orthonormality conditions (6.58) are trivially satisfied.

Due to errors in relative rotations, the individual quaternions in the solution vector have different lengths. Because of this, each ij-constraint, i.e. the four equations (6.60) demanded by the ij-relative rotation, has a different influence (weight), which is approximately proportional to the lengths of the resulting i- and j-quaternions. The shorter are the two four-vectors, the smaller attention has to be given to the four equations. As a consequence, the difficult partial reconstructions, i.e. those which significantly differ from the remaining ones, are given small attention. They get weighted down to better fit the majority of constraints.

Remark. A solution would be to add the constraint on unit lengths of all resulting quaternions:

$$(r_0^i)^2 + (r_x^i)^2 + (r_y^i)^2 + (r_z^i)^2 = 1 \qquad \text{for all } i \tag{6.61}$$

Unfortunately, sofar no satisfactory way for solving a linear system like (6.60) with quadratic constraints like (6.61) is known.

Note that rotation registration using quaternions has been proposed before us without our knowledge by Govindu in [26].

6.8.2 Registration using Approximate Rotations

An alternative way is to solve system (6.57) without satisfying orthonormality constraints (6.58). In fact, system (6.57) consists of three smaller subsystems

$$\mathbf{r}_k^j - \mathtt{R}^{ij}\mathbf{r}_k^i = 0_{3\times 1} \qquad \text{for all } ij \tag{6.62}$$

for $k = 1, 2, 3$, where $\mathbf{r}_k^i$ are columns of $\mathtt{R}^i$, $\mathtt{R}^i = [\mathbf{r}_1^i \mathbf{r}_2^i \mathbf{r}_3^i]$. The solution for approximate rotations can be found as the best three linearly independent least squares solutions to system (6.62). System (6.62) is sparse and thus can be solved, e.g., using MATLAB's EIGS. See [65] (section 6.3.1) for details on a solution to a similar system. The orthonormality constraints (6.58) are enforced by projecting the approximate rotation to the closest rotation in the Frobenius norm using SVD [66] (section 6.7.2).

Compared to [66], no auxiliary variables rotating the partial reconstructions to the global coordinate system are needed. Thus, this solution is simpler and faster. We observed that it is also more stable.

Results got improved when ij-equations (6.62) corresponding to the ij-EG were reweighted by $\min(a, 400)$, where a is the number of inliers in the ij-EG. Solution to (6.62) can be found very efficiently. Rotation registration of 259 views using 2049 relative rotations in the Tête scene (see figure 6.27) using MATLAB's EIGS took only 0.37 seconds.

Comparison with Quaternions

On the Head scene [66] (see figures 3.3 and 6.21), the ratio between the maximum and minimum quaternion lengths from (6.60) was 5.04. On the other hand, the norms of the 3×3 matrices found by (6.62) were very close to each other (less than 1%). Norms of individual 3-vectors were even closer (less than 0.1%). The maximum Frobenius norm of the difference between the relative rotation and the relative rotation after registration, $||\mathtt{R}^{ij} - \mathtt{R}^j \mathtt{R}^{i\top}||$, was 1.98 and 0.37 for quaternions and approximate rotations, respectively. The fact that the first number is very

close to the maximum possible norm (which is 2) shows that the method using quaternions in not usable in practice. In the rest of the section, only approximate rotations are used.

The reason why the least squares solution is worse for quaternions than for approximate rotations is perhaps the following. When searching for the most suitable rotations, it is easier to search in the space of approximate rotations (all 3×3 matrices) than in the space of rotations (quaternions). The latter is a small manifold included in the first space. In both cases, a solution that well satisfies all constraints on relative rotations is searched for.

The inconsistencies in constraints prove as (i) getting off the manifold and (ii) changing lengths of vectors representing individual rotations. The approximate rotations "use" both (i) and (ii) "effects" and thus are in higher accordance with all constraints as they can be off the manifold. (It is not far from the manifold, as will be shown on experiments.) For quaternions, (i) is not possible. This thus causes a bigger pressure on (ii), i.e. deformation of quaternion lengths. As a consequence, the constraints with very short quaternions are given a very low attention, which is the undesired side effect. This effect happens with approximate rotations as well but with differences of lower order magnitudes, as shown above.

There is an alternative explanation why approximate rotations behave better than quaternions. In the absence of noise in the data, proposition 2, *p*57 guarantees orthogonality of columns of the 3×3 matrices provided that the three solutions to system (6.62) are linearly independent. When noise is present, columns of the 3×3 matrices are only close to orthogonal according to an "approximate version" of proposition 2.

6.8.3 Data Compression and Clarification

We found out that it is possible to represent each partial reconstruction using four points only while capturing the overall geometry well. The idea comes from projective factorization using perspective cameras [113]. Projection matrices of a partial reconstruction, $\mathtt{P}$, multiplied with all points reconstructed in that partial reconstruction, $\mathbf{X}$, form so-called *rescaled measurement matrix* $\lambda\mathbf{x} = \mathtt{P}\mathbf{X}$, where the measured image points $\mathbf{x}$ are rescaled by depths λ element-wise, $\lambda_p^i \mathbf{x}_p^i = \mathtt{P}^i \mathbf{X}_p$ [113]. Here we work with projected points $\mathtt{P}\mathbf{X}$ instead of the rescaled measured image points $\lambda\mathbf{x}$. It is equivalent when there is no noise in the data. Usage of the projected points has the advantage that the rescaled measurement matrix is less affected by noise when cameras are well estimated (which is often the case).

The desired four points are chosen so that the corresponding four columns in $\mathtt{P}\mathbf{X}$ represent the four dimensional subspace spanned by all columns of $\mathtt{P}\mathbf{X}$. Thus, the necessary condition is that the chosen four columns are linearly independent. There are many such quadruplets, therefore an additional criterion is needed. Before formulating it, a criterion for identifying mismatches will be given.

Identifying Mismatches

True matches connect one or several surfaces visible in an image pair. True matches connecting the same surface are (i) localized close to one another in the images and (ii) have similar depths. As a result, true matches form clusters in the rescaled image space while mismatches are far from the remaining data due to incorrect depths. To ensure that the clusters are formed, the images of the scene must contain sufficiently large surfaces on which multiple matches forming a cluster could be detected and matched. There are scenes which do not satisfy this assumption like, e.g., many tiny branches of a tree. However, such scenes would hardly be matched by any algorithm, thus the assumption on scenes containing sufficiently large surfaces is not so restrictive in practice. Any clustering algorithm could be used to find individual surfaces corresponding to the clusters. Matches contained in no or small clusters could be thrown away as most likely

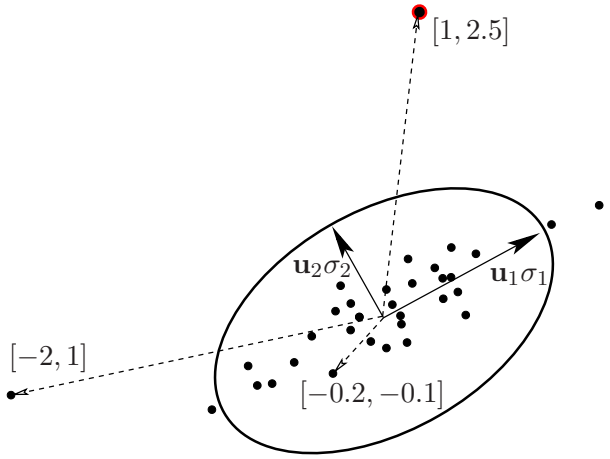


Figure 6.23: Each point represents a vector in a two-dimensional vector space. The ellipse characterizes the fitted Gaussian to the centered data. The ellipse center is in $[0, 0]$ and its half-axes are $\mathbf{u}_1\sigma_1$ and $\mathbf{u}_2\sigma_2$ where $[\mathbf{u}_1\mathbf{u}_2]\,\mathrm{diag}(\sigma_1, \sigma_2)\mathtt{V}^\top$ is the "economy size" SVD factorization of $\mathtt{P}\mathbf{X}$. It is drawn for the 2D case instead of 6D. The ellipse shape characterizes the most of the data mass. The most likely (ML) mismatches are the most distant points from the center of the Gaussian in the Mahalanobis distance given by the auto-covariance matrix of the Gaussian. The coordinates are drawn at three points. These are also rows of the $\mathtt{V}$ matrix. Although the leftmost point is the most distant from $[0, 0]$, the upmost point is a more likely mismatch as its distance is larger in the ellipse coordinate system: $||[1, 2.5]|| > ||[-2, 1]||$.

(ML) mismatches. Nevertheless, in this work we did something much simpler and that clustering with one cluster only.

In this work, the main purpose was to reliably remove all mismatches as the L_∞-norm, i.e. the maximum reprojection error, is minimized in translation estimation [44], which may be hundreds of pixels due to a single mismatch. Thus, to get a reasonable estimate using [44], all (or at least most) mismatches have to be removed. We observed on the presented scenes that either an EG was non-existent or its inliers were contaminated by a low amount (less than $\epsilon = 25\%$) of mismatches. When more mismatches are present, such EG is likely to be detected and removed after translation registration, see section 6.8.5. A Gaussian was fitted to the data in the rescaled image space and a prescribed amount, ϵ, of most distant points was thrown away as the ML mismatches, see figure 6.23. Localizing the largest cluster (or a set of large clusters) by a single Gaussian is justifiable when the inter-cluster distances are relatively small compared to the distances to mismatches. This simple way worked well on scenes presented here and many others.

After estimating the data mean and subtracting it from all vectors, the covariance matrix of the Gaussian is obtained using SVD. The ML mismatch is the most distant point in the coordinate system given by the Gaussian covariance matrix. Its corresponding row in matrix $\mathtt{V} \in \mathbb{R}^{n\times 4}$ has the largest norm, where $\mathtt{P}\mathbf{X} = \mathtt{U}\,\mathrm{diag}(\sigma_1, ..., \sigma_4)\mathtt{V}^\top$ is the "economy size" SVD decomposition. It is illustrated on figure 6.23, see the explanation there.

All ϵ ML mismatches can be either (i) removed at once or (ii) one by one while refitting the Gaussian after each ML mismatch removal from $\mathtt{P}\mathbf{X}$. The latter way was used in this work as a higher stability can be expected. The SVD decomposition can be carried out efficiently, see lines 2–4 of algorithm 6. Note that the most time consuming operation is SVD applied to a

```
for k = 4:-1:1
  [U,s,v] = svd(R*R',0);                          % svd of a long matrix
  S = sqrt(diag(s(1:k,1:k)));                     % using svd of a short one
  V = ((diag(1./S)*v(:,1:k)')*R)';                % R = U*diag(S)*V'
  len = V'.^2; if k > 1,
    len = sum(len); end                           % squared lengths of rows of V
  best = find(len == max(len));
  p(k) = best(1);
  C = R(:,p(k));                                  % the chosen column
  R = R - C*(pinv(C)*R);                          % subtract its span
end
```

Algorithm 6: Choosing the four most different points representing a partial reconstruction. In MATLAB code, variable R contains the rescaled measurement matrix, $\mathtt{P}\mathbf{X}$. Indices of the chosen four points are stored in variable p.

Figure 6.24: Image pair 19-22 in the Raglan scene. Points satisfying EG of this image pair (top row). Non-mismatch candidates identified before the multiview registration (bottom left). The four points used for translation registration (bottom right).

6×6 matrix irrespective of the number of points.[40] An example of identified ML mismatches at $\epsilon = 25\%$ is shown in figure 6.24.

Normalization. As the procedure is done on the rescaled measurement matrix, i.e. on rescaled image data, the image coordinates should be normalized to be close to one [31] and the resulting $\mathtt{P}\mathbf{X}$ should be balanced by rescaling its columns and row triplets, as described in [113].

ML mismatches are identified prior to rotation registration. Doing it afterwards based on

[40] If needed, even a more efficient implementation is possible using incremental SVD [10] instead of the standard SVD by exploiting the fact that only four instead of six basis vectors are needed to span the space generated by the columns of the $\mathtt{P}\mathbf{X}$ matrix.

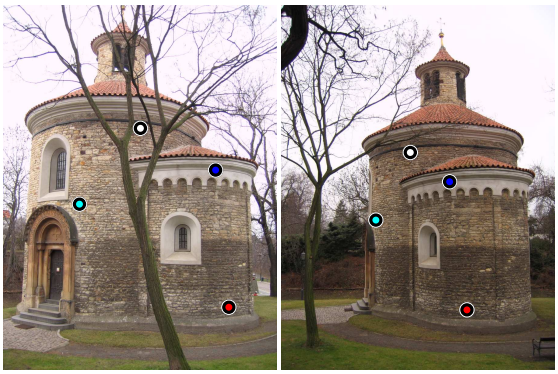


Figure 6.25: Four most different points chosen after the removal of $\epsilon = 25\%$ ML mismatches. Image pair 41-48 in the St. Martin rotunda is shown. The points lie in different depths and thus capture the 3D geometry of the image pair well.

the partial reconstruction reestimated using the registered rotations might be incorrect as the estimate of the registered rotations may be severely corrupted due to non-existent EGs (see section 6.8.5).

As a side effect of the removal of all ϵ ML mismatches, many true matches are removed as well. Nevertheless, it is not a problem as the left data constrain the multiview reconstruction sufficiently, as will be shown in section 6.8.4.

Reconstruction Represented by Four Points

After $\mathtt{P}\mathbf{X}$ has been cleared of mismatches, the same Gaussian fitting technique is used for choosing the four points for representation of the partial reconstruction. If the data contains a mismatch, the most different point is the ML mismatch. However, after the data was cleared of mismatches, the most different point is the best inlier for representing the geometry. The four points are found in the following way. After identifying the most different point, the whole matrix is projected onto the span of the chosen column and subsequently subtracted from $\mathtt{P}\mathbf{X}$. This is repeated four times. The procedure is summarized in algorithm 6.

The chosen points lie in different depths as well as the ML mismatch does. However, here it is advantageous as the different depths capture the 3D geometry of the two images well, as can be seen in figure 6.25. Note that if the data contained any previously not removed mismatch, it would very likely appear among the chosen four points.

Remark. The ML mismatch identification and the choice of the four representative points work for projective reconstruction as well since product $\mathtt{P}\mathbf{X}$ depends on images and not on the choice of a reference frame. The depths do not have to be positive, nor the cameras calibrated. The only thing that matters is how the columns corresponding to points are situated in the subspace generated by normalized columns of $\mathtt{P}\mathbf{X}$ (cf. the note on normalization above).

6.8.4 Translation Registration

In [66] (section 6.7), translations and points in each partial reconstruction were estimated using [44]. Then, all partial reconstructions were refined together using bundle adjustment (BA) while keeping rotations registered. Unlike in [66], in this work, no such intermediate BA is performed. The reason is that the precision of the presented rotation registration is satisfactory when combined with the robust point sampling explained above.

Figure 6.26: The Raglan scene. An overall view with a bridge (left) and a view inside from the other side (right).

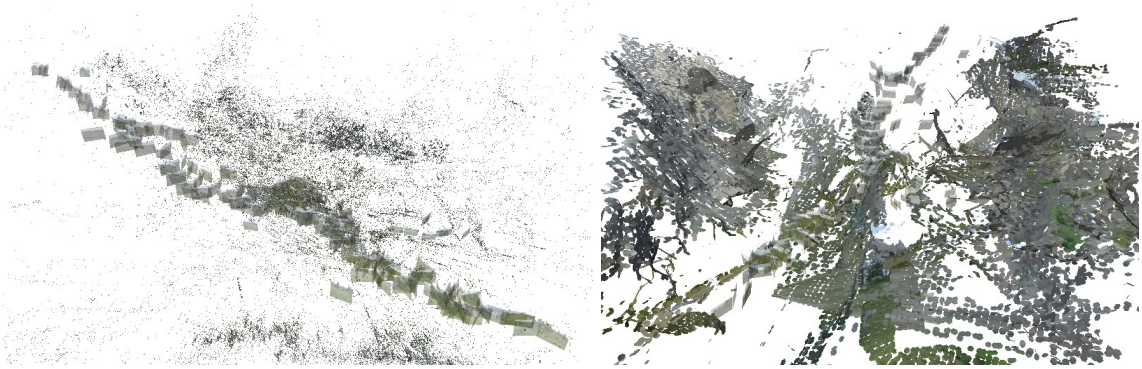


Figure 6.27: The Tête scene. The profile view - top of the mountain is on the left-hand side (left). View from the valley up (right).

Method [44] is applied only once on the data from all partial reconstructions. However, each partial reconstruction is represented by four points chosen as explained in section 6.8.3 instead of almost all points. Thus, it is much faster. After translation registration, BA on all data was done and dense reconstructions were obtained using [17, 12].

The Raglan scene [101] was captured on 46 images, 238 EGs were found (see details in section 6.8.6). When [44] was applied on all points in all partial reconstructions (186131 points in total), the maximum residual of 98.57 pixels was obtained in 3 hours and 6 minutes. When using only the four representative points, the maximum residual of 98.46 pixels was obtained in 4.68 seconds. This demonstrates that the four points represent geometry of the individual reconstructions well while achieving a huge speedup (of factor 2385 at this particular scene). When using quadruplets chosen from the non-mismatch candidates at $\epsilon = 25\%$, the obtained error decreased to 22.30 pixels. It was manually verified on several quadruplets with largest residuals [106] that none of them included a mismatch, although there were many in the data, see figure 6.24. When using the intermediate BA with rotations kept registered before applying [44] on all image pairs, the maximum error dropped to 12.09 pixels. The reconstruction is shown in figure 6.26.

The Tête de Plate Longe (shortly Tête) scene (259 images, 2049 EGs) was reconstructed with the largest residual of 38 pixels in 74 seconds. Manual inspection verified that no mismatch

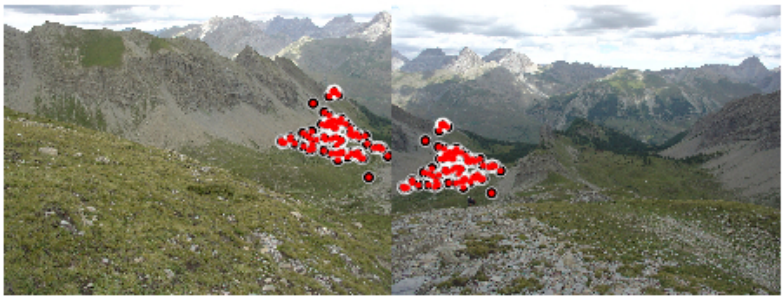


Figure 6.28: The Tête scene. View from top into the left (left) and right valley (right).

was present. We tried several strategies for reweighting equations (6.62) based on residuals in individual partial reconstructions, however no general strategy was found (the best trial dropped to 27). See figures 6.27 and 6.28.

6.8.5 Robust Rotation Estimation

It turned out that even if the found relative rotation is close to the desired one in the Frobenius norm, i.e. $||\mathtt{R}^{ij} - \mathtt{R}^{j}\mathtt{R}^{i\top}||$ is small, the partial reconstruction with rotation replaced by the found rotation (and with translations reestimated by [44]) may still produce large residua. This effect could be reduced by using rotation uncertainties [105]. However, a more serious problem is when the rotation registration is contaminated by some non-existent EG. Fortunately, it has been observed that the points from such an EG have large residua after the rotation and translation registration. Thus, it is straightforward to remove such partial reconstruction and reestimate rotations and translations.

As mentioned in the introduction, it was proved [106] for a wide class of L_∞ problems that the set of measurements with the greatest residual must contain at least one outlier. However, it is not the case of the least squares rotation estimate presented here. Nevertheless, it will be shown on two scenes that this property holds in practice even for the L_∞ problem [44] initiated by the least squares solution to (6.62).

Our least squares solution to (6.62) provides quite a good estimate even when many relative rotations came from non-existent EGs (in the Zwinger scene, more than 156 (8%) EGs were non-existent). The reason why it works so well is perhaps that the existent EGs support each other while the non-existent ones rather do not as they raised almost randomly and independently. However, each non-existent EG deteriorates the quality of the solution.

Unlike [106], we do not remove single points but whole partial reconstructions, in which some of the four points reached the maximum residual. This brings an additional speedup besides the compression to four points.

The St. Martin rotunda (124 images, 1670 EGs) was reconstructed with the mean/maximum residual of 1.5/7.66 pixels after 11 iterations of removing EGs with the largest residual and rotation and translation reestimation, see figure 6.29. There were 13 non-existent EGs detected (manually checked), one of which is shown in figure 6.29. In some iterations, more EGs with the same maximum residual (at some of the four points) were removed. The dense reconstruction using [17, 12] in figure 6.30 demonstrates that the presented method reaches a high precision. The surface parts from different views shown in different colors due to varying lightning conditions fluently connect to each other.

On the Zwinger scene (199 images, 1954 EGs), method [44] produced error of 229 pixels in 51 seconds. There were many non-existent EGs, see figure 6.29. It seems that after the maximum

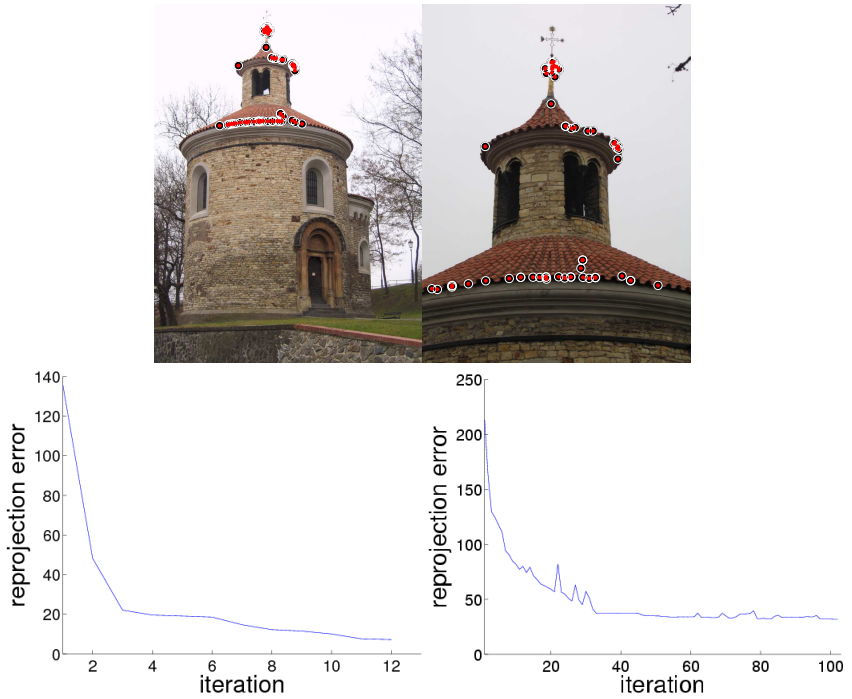


Figure 6.29: Iterative removal of EGs with the largest residual. One of 13 non-existent EGs in the St. Martin rotunda: image pair 4-119 (top row). Decreasing of the maximum residual is shown for the St. Martin (bottom left) and the Zwinger scene (bottom right).

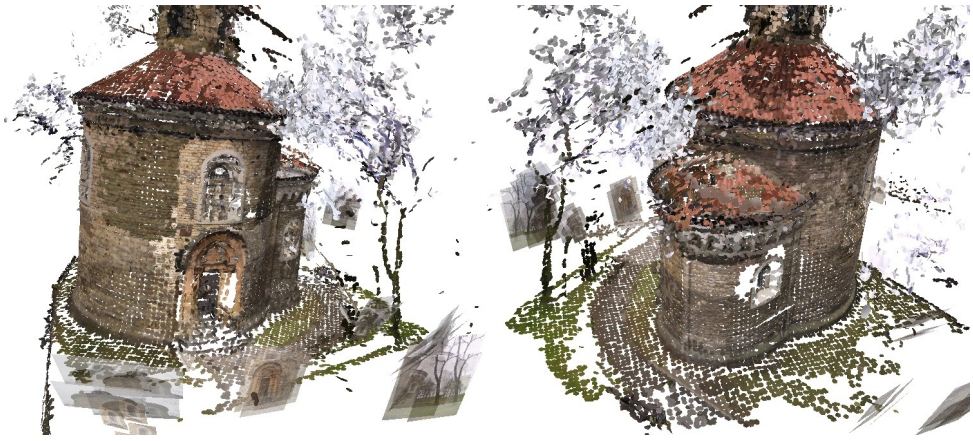


Figure 6.30: The St. Martin rotunda. Front and back view of the dense reconstruction with some cameras shown as image planes. Note the details as the tree and the footpath around the building. The clouds come from tiny branches.

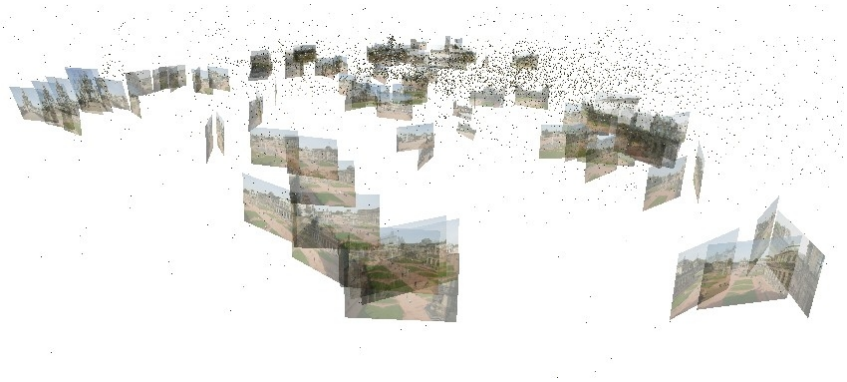


Figure 6.31: The Zwinger scene. Only quadruplets of points representing EGs are shown. Non-existent EGs with many mismatches between repetitive structures on building facades are still present.

residual dropped below 35 pixels (at iteration 51, 123 EGs removed), it was hard to improve the precision more. The reconstruction shown in figure 6.31 was done using the result of iteration 100 (156 EGs removed, error 31 pixels). It turned out after manual inspection that still some non-existent EGs remained in the data.

6.8.6 Experiments

In experiments reported here, pairwise image matching was done with Local Affine Frames [70] constructed on intensity and saturation MSER regions, LaplaceAffine and HessianAffine [75] interest points. Additional matches were found using SIFT features [56]. Only some image pairs were matched on the large Tête and Zwinger scenes. See section 5.1 for details on the used heuristic.

The six-point RANSAC [111] with plane detection [16] was run on the matched pairs and the focal length was calibrated as the mean of all estimates (see section 5.3). Then, BA on all pairs with focal lengths kept equal (but varying) was run, followed by the five-point RANSAC [77] and track merging. Radial distortion was not removed from the images.

Due to a few repetitive structures in the Raglan scene and a huge amount of them in the Zwinger scene, RANSAC on many-to-many correspondences had to be used, details can be found in section 4.3.

It was desired to forbid all pairs not suitable for dense stereo. These are especially pairs with (nearly) coinciding camera centers forming a panorama. If some pair should fit a panorama model, it must fit a weaker homography model at least so well. Thus, only pairs with 90% inliers lying on a (dominant) plane need to be checked for being a panorama, the remaining ones cannot be a panorama. Fitting the panorama model was started by making the two camera centers coincident by setting them to their mean. Then BA constrained to keep the camera centers equal was run. Many panoramas were successfully detected but some not, which can be seen on the Tête scene in figure 6.27right. See section 5.5 for a better algorithm for panorama detection.

Summary and Conclusions

A practical method for automatic reconstruction was presented. It was shown to work on hundreds of images. 99.68% of the measurement matrix of the Tête scene were missing due to occlusions. There is no chance for any factorization method to deal with such a large amount of

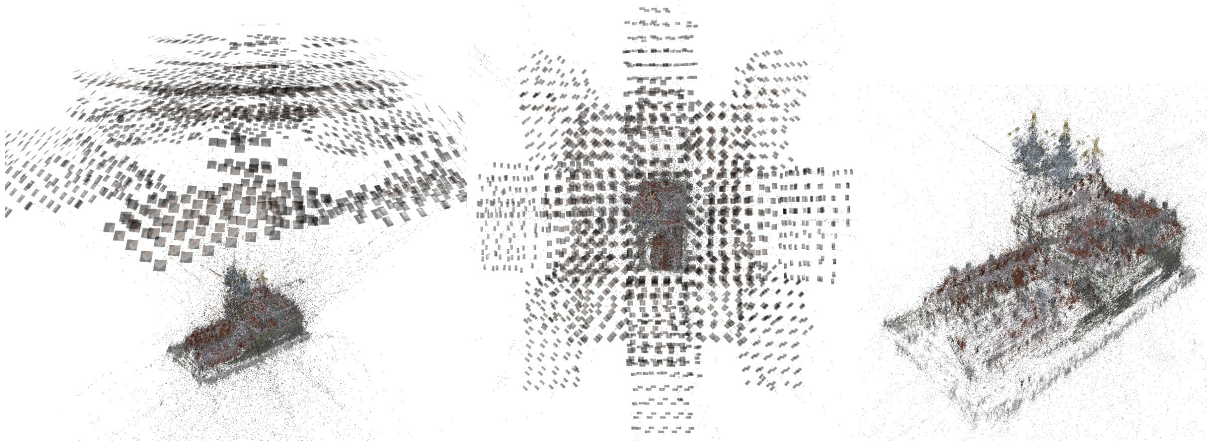


Figure 6.32: Part of a city paper model captured on 2126 images. Front-side view (left), top view (middle), and detailed view on buildings (right). 660037 sparse 3D points were reconstructed, 99.85% data of the MM are missing. Due to the data size, the point cloud from dense stereos could not fit into 16GB memory.

missing data. The whole algorithm uses only two-view correspondences except for the final BA, which starts with quite low errors (from 7 to 30 pixels) and thus can change the overall geometry only slightly. This means that the overall geometry is mostly determined by the rotation and translation registration. The rotation registration takes a fraction of a second on hundreds of images and the translation registration takes around a minute. Both should be repeated when the data is contaminated by non-existent EGs. Even in this case, the total running time is in the order of minutes, which is a fraction of the time spent by BA in the incremental structure from motion (SfM) [108]. Note that images of the presented scenes are very sparsely captured compared to [108].

Closed image sequence is a problem for any incremental SfM as the first and the last camera positions get misaligned. In our approach, using all EGs at once has the advantage that many closed loops among images can be handled.

It has been shown that the presented method is robust to some contamination by non-existent EGs. The contamination in the Zwinger scene is an extreme one: hundreds of non-existent EGs, most of the existent EGs have mismatches on repetitive structures. To reconstruct this scene better, detecting non-existent EGs prior to rotation registration seems to be needed.

More reconstructed scenes can be seen at [3]. After publishing [67], the method was tested on even larger data with more than 2000 images, see figure 6.32.

7 Multiview Reconstruction for Lines

This chapter deals with reconstruction of lines from perspective images. In section 7.1, projective factorization for lines for perspective cameras is presented [64]. The main achievement is the problem formulation. The presented algorithm is not very practical, because it is not robust to noise in line measurements as no geometrically meaningful error is minimized. Moreover, it cannot deal with occlusions. It turns out that knowledge of internal camera calibration can help in reconstructing lines as everything can be done in the metric space instead of the projective one. An algorithm exploiting calibrated cameras is proposed in section 7.2. It can be described as metric gluing of three-view reconstructions of lines.

7.1 Factorization with Perspective Cameras

This section[1] presents a method for line reconstruction from many perspective images by factorization of a matrix containing line correspondences. No point correspondences are used. We formulate the reconstruction from line correspondences in the language of Plücker line coordinates. The reconstruction is posed as the factorization of $3m \times n$ matrix $\mathtt{S}$ into the product $\mathtt{S} = \mathtt{QL}$ of $3m \times 6$ projection matrix $\mathtt{Q}$ and $6 \times n$ line matrix $\mathtt{L}$, both satisfying the Klein identities, see section 2.2. The $\mathtt{S}$ matrix contains coordinates of lines detected in perspective images. Similarly to reconstruction from point correspondences in perspective images, matrix $\mathtt{S}$ has to be properly rescaled before it can be factorized. We present a scaling of image line coordinates based on trifocal tensors that is analogical to the scaling proposed by Sturm and Triggs for points. We present an SVD based factorization enforcing the Klein identities on $\mathtt{Q}$ and $\mathtt{L}$ in a noise-free situation. We show experiments on real data that suggest that a good reconstruction may be obtained even if data is noisy and the identities are not enforced exactly. We also discuss an extension of the method for images with occlusions.

In this section we formulate the reconstruction from line correspondences by factorization in a perspective setup. We concentrate on giving the formulation of the problem and on demonstrating in experiments that meaningful results are obtained. We show that a suitable formulation leads to a well posed problem that can be solved even in the presence of noise without solving every step optimally in full generality. We concentrate here only on the situation when there is no occlusion in the scene. An extension towards occlusions is possible but will be presented elsewhere. Foundations for our method follow.

First, we use Plücker coordinates to represent lines in three dimensional space as well as in images. Plücker coordinates of lines in space are *linearly* projected to Plücker coordinates of lines in images, exactly the same way as homogeneous 3D point coordinates are projected linearly into homogeneous 2D point coordinates. Only thanks to the linearity of the projection, a factorization is possible.

Second, a scaling method has to be proposed in order to properly scale image line coordinates in the line measurement matrix. We present a line scaling technique that exploits trifocal tensors

[1] Most of this section was published in [64]. Andrew Zisserman from the University of Oxford kindly provided the House data, Tomáš Werner from the Czech Technical University in Prague provided the automatic line matches in the House scene and the routine for the line bundle adjustment, and Martin Urban from the Czech Technical University in Prague provided the code for step 2 in algorithm 7.

of line correspondences across triplets of views.

Third, the representation of lines by Plücker coordinates requires that the elements of a six dimensional Plücker vector satisfy a quadratic identity to represent a line in space. Thus, it is not enough to factorize the line measurement matrix by a simple SVD-based factorization but it is necessary to do it so that the reconstructed representation of structure and motion satisfy all necessary identities. We show how to achieve it for a noise free data by a simple transformation of an SVD-based factorization into the coordinate system of a reconstruction from one triplet of images.

Finally, everything becomes more difficult when noise is present in data as it is more difficult to enforce the required identities and to obtain a consistent representation in an optimal way. We do not show how to do it optimally here but we show that even a simple, and probably not very optimal, technique provided a meaningful reconstructions. This technique can be used to initialize a non-linear bundle adjustment. We believe that there is a good reason to hope that much better results would be obtained when employing better estimation techniques to cope with noise.

The principal difference of this work to Triggs' factorization method on lines [125] is in representation of lines. Method [125] represents a line by a pair of points which are transfered from the first into other images using the epipolar geometries known from the trifocal tensors (so-called point transfer). The advantage of [125] is that both points and lines can be used. The price for not doing the point transfer in our method is payed by the necessity for enforcing the nonlinear identities.

The work is structured as follows. In section 2.2, *p*7 line representation by Plücker coordinates is adopted and the factorization based reconstruction from line correspondences in perspective images is formulated. The main idea of the approach is spelled out below and the two key components of the approach, the scale factors estimation and the enforcing the Klein identities, are described in more detail. Experimental results follow and further extensions are discussed in the concluding section.

The Main Idea of the Factorization Algorithm

A factorization method requires that all elements of the $\mathtt{S}$ matrix (*p*8) are known. Therefore, the scale factors in $\mathtt{S}$ have to be estimated before $\mathtt{S}$ can be factorized. The necessary condition for $\mathtt{S}$ being factorizable is that it is of rank six. However, it is not a sufficient condition and the matrices resulting from a plain factorization into a product of $\hat{\mathtt{Q}} \in \mathbb{R}^{3m\times 6}$ and $\hat{\mathtt{L}} \in \mathbb{R}^{6\times n}$ by SVD as in equations (7.6) and (7.7) do not have to satisfy the Klein identities (2.6) on rows of $\hat{\mathtt{Q}}$ and columns of $\hat{\mathtt{L}}$, respectively. The situation can be remedied by finding a projection of the respective vectors onto the Klein quadric.

If $\mathtt{S}$ is a valid rescaled line measurement matrix, then $\mathtt{Q}$ and $\mathtt{L}$ satisfying (2.6) exist. Matrices $\mathtt{Q}$ and $\mathtt{L}$ can always be obtained from $\hat{\mathtt{Q}}$ and $\hat{\mathtt{L}}$ by some transformation $\mathtt{H} \in \mathbb{R}^{6\times 6}$, $\operatorname{rank}\mathtt{H} = 6$, as $\mathtt{Q} = \hat{\mathtt{Q}}\mathtt{H}$ and $\mathtt{L} = \mathtt{H}^{-1}\hat{\mathtt{L}}$ for each factorization pair is bound by a change of the basis

$$\mathtt{S} = \underbrace{\hat{\mathtt{Q}}\mathtt{H}}_{\mathtt{Q}}\underbrace{\mathtt{H}^{-1}\hat{\mathtt{L}}}_{\mathtt{L}}. \tag{7.1}$$

Estimating the Scale Factors

The scale factor estimation is done by computing partial reconstructions from triplets of images using trifocal tensors and re-projecting the reconstructions back into the images. Scale factors are computed from the difference between the reprojected and the original image lines. Equation (7.2) is the best solution for γ_l^i in the least squares sense, which is a variation on equation (3)

1. Estimate the trifocal tensor $\mathcal{T}$ using line correspondences [30].
2. Compute proj. matrices $\mathtt{P}$, $\mathtt{P}'$, and $\mathtt{P}''$ using $\mathcal{T}$ [30, 126].
3. Compute proj. matrices $\mathtt{Q}$, $\mathtt{Q}'$, and $\mathtt{Q}''$ using equation (2.4).
4. Compute $\mathbf{L}_l$ as intersections of the back-projected image lines [30] i.e. assemble the matrix $\mathtt{W}$

$$\mathtt{W} = \begin{bmatrix} \mathbf{l}^\top \mathtt{P} \\ \mathbf{l}'^\top \mathtt{P}' \\ \mathbf{l}''^\top \mathtt{P}'' \end{bmatrix}$$

   and set $\mathbf{L}_l = \mathtt{v}(:,3) \vee \mathtt{v}(:,4)$ where $[\mathtt{u}, \mathtt{s}, \mathtt{v}] = \mathrm{SVD}(\mathtt{W})$ and $\mathbf{X}_1 \vee \mathbf{X}_2$ is a join of 3-space points $\mathbf{X}_1$ and $\mathbf{X}_2$ into the line in Plücker coordinates [30].
5. Estimate scale factors γ_l, γ'_l, and γ''_l according to

$$\gamma_l^i = \frac{\bar{\mathbf{l}}_l^i \cdot \mathbf{l}_l^i}{\|\mathbf{l}_l^i\|^2} \tag{7.2}$$

   where $\bar{\mathbf{l}}_l^i$ are the projected lines $\mathbf{L}_l$ into the image i, $\bar{\mathbf{l}}_l^i = \mathtt{Q}^i \mathbf{L}_l$.

Algorithm 7: Scale factor estimation from a triplet of views.

in [113].[2] The whole algorithm for scale factor estimation is summarized in algorithm 7.

Scale factors resulting from independent partial reconstructions from triplets of views may be mutually inconsistent, which means that scale factors of a given image line, resulting from different triplets of views, differ. Therefore triplets of views must be established so that they overlap by at least one view and in the sense that such overlaps form one connected component. Rescaling equations (2.3) of all triplets of views can be then chained together for any given line l over m views by column rescaling to give a consistent $(\gamma_l^1, \gamma_l^2, \ldots, \gamma_l^m)^\top$ [96].

Enforcing the Klein Quadric Identities

The Klein quadric identities (2.6) on rows of $\mathtt{Q}$, resp. columns of $\mathtt{L}$, can be written in a matrix form as

$$\mathrm{diag}(\mathtt{Q}\mathtt{E}_\mathtt{r}\mathtt{Q}^\top) = \mathtt{0}_{3m\times 1}, \quad \text{resp.} \quad \mathrm{diag}(\mathtt{L}^\top \mathtt{E}_\mathtt{r} \mathtt{L}) = \mathtt{0}_{n\times 1}, \tag{7.3}$$

where the 6×6 matrix $\mathtt{E}_\mathtt{r}$ is of the form

$$\mathtt{E}_\mathtt{r} = \begin{bmatrix} \mathtt{0}_{3\times 3} & \mathtt{I}_{3\times 3} \\ \mathtt{I}_{3\times 3} & \mathtt{0}_{3\times 3} \end{bmatrix}$$

and $\mathtt{I}_{3\times 3}$ is the identity matrix. Equations (7.3), after applying (7.1), become

$$\begin{aligned} \mathrm{diag}(\hat{\mathtt{Q}}\mathtt{H}\mathtt{E}_\mathtt{r}\mathtt{H}^\top \hat{\mathtt{Q}}^\top) &= \mathtt{0}_{3m\times 1}, \quad \text{resp.} \\ \mathrm{diag}(\hat{\mathtt{L}}^\top \mathtt{H}^{-\top} \mathtt{E}_\mathtt{r} \mathtt{H}^{-1} \hat{\mathtt{L}}) &= \mathtt{0}_{n\times 1}, \end{aligned} \tag{7.4}$$

which is quadratic in terms of elements of $\mathtt{H}$ and could be solved only nonlinearly. Fortunately, $\mathtt{H}$ can be found by another way. The following holds.

[2] As pointed out by one of the reviewers of [64], the scale factors can be estimated without explicit reconstruction (which is, however, needed in section 7.1) in the following way. Given consistent scales for the epipoles (in the sense of [113, 125, 124]), a consistent scaling for the image lines is $\gamma^i \mathbf{l}^i \cdot \mathbf{e}^{ij} = -\gamma^j \mathbf{l}^j \cdot \mathbf{e}^{ji}$ for any i, j.

1. Establish triplets of views among m views such that the triplets overlap as explained in section 7.1. For each triplet of views, compute the scale factors using algorithm 7.
2. Chain the rescaling equations (2.3) of all triplets of views together for any given line l over m views to give a consistent $(\gamma_l^1, \gamma_l^2, \ldots, \gamma_l^m)^\top$. Denote the triplet whose scale factors have not been changed during the rescaling by $\mathbf{t}$.
3. Factorize complete rescaled line measurement matrix $\mathtt{S} = \left[\, \gamma_l^i \mathbf{l}_l^i \,\right]_{i=1\ldots m, l=1\ldots n}$ into matrices $\hat{\mathtt{Q}}$ and $\hat{\mathtt{L}}$ as
$$\hat{\mathtt{Q}} = \mathtt{u}(:, 1:6) \tag{7.6}$$
$$\hat{\mathtt{L}} = \mathtt{s}(1:6, 1:6)\ \mathtt{v}(:, 1:6)^\top \tag{7.7}$$
where $[\mathtt{u}, \mathtt{s}, \mathtt{v}] = \mathrm{SVD}(\mathtt{S})$.
4. Find the transformation matrix $\mathtt{H}$ transforming the rows of $\hat{\mathtt{Q}}$ corresponding to triplet $\mathbf{t}$ into the basis of the partial reconstruction of triplet $\mathbf{t}$, i.e.
$$\begin{bmatrix} \hat{\mathtt{Q}}^i \\ \hat{\mathtt{Q}}^j \\ \hat{\mathtt{Q}}^k \end{bmatrix} \mathtt{H} = \begin{bmatrix} \mathtt{Q} \\ \mathtt{Q}' \\ \mathtt{Q}'' \end{bmatrix}$$
where $\mathtt{Q}$, $\mathtt{Q}'$, and $\mathtt{Q}''$ come from scale factor estimation for triplet $\mathbf{t} = (i, j, k)^\top$, step 3 in algorithm 7.[a]
5. Apply transformation $\mathtt{H}$ so that the result is close to the Klein quadric: $\tilde{\mathtt{Q}} = \hat{\mathtt{Q}}\mathtt{H}$, $\tilde{\mathtt{L}} = \mathtt{H}^{-1}\hat{\mathtt{L}}$.
6. Project rows of $\tilde{\mathtt{Q}}$ and columns of $\tilde{\mathtt{L}}$ onto the Klein quadric as in section 7.1 to gain $\mathtt{Q}$ and $\mathtt{L}$, respectively.

[a] An alternative way is to find $\mathtt{H}$ as $\mathtt{H} = \mathtt{G}^{-1}$ where $\hat{\mathtt{L}}^\top \mathtt{G} = \mathtt{L}^{\mathbf{t}\top}$, which appeared to be less sensitive to noise.

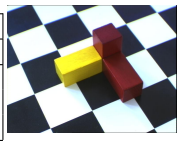
Algorithm 8: Scene reconstruction from lines.

Proposition 7 *Let* $\mathtt{S}$ *be a rescaled line measurement matrix (no noise in data) composed from non-degenerate perspective images of lines in a scene taken by cameras in general positions. Let* $\mathtt{S} = \hat{\mathtt{Q}}\hat{\mathtt{L}}$ *be a plain factorization of* $\mathtt{S}$ *by* SVD *(not necessarily satisfying Klein identities (7.3)). Let*
$$\mathtt{S}_{9\times n} = \dot{\mathtt{Q}}_{9\times 6}\dot{\mathtt{L}}$$
be a partial reconstruction of the lines in the scene from a triplet $\mathbf{t}$ *of images obtained by algorithm 7 (i.e.* $\dot{\mathtt{Q}}_{9\times 6}$ *and* $\dot{\mathtt{L}}$ *satisfy the Klein identities (7.3)). If matrix* $\mathtt{H}$ *satisfies*
$$\hat{\mathtt{Q}}_{9\times 6}\mathtt{H} = \dot{\mathtt{Q}}_{9\times 6} \tag{7.5}$$
then $\mathtt{Q} = \hat{\mathtt{Q}}\mathtt{H}$ *as well as* $\mathtt{L} = \mathtt{H}^{-1}\hat{\mathtt{L}}$ *satisfy the Klein identities (7.3).*

Proof: Columns of $\dot{\mathtt{L}}$ satisfy Klein identities since they are equal to coordinates of space lines reconstructed from a triplet of images via a trifocal tensor. $\mathtt{L} = \mathtt{H}^{-1}\hat{\mathtt{L}} = \dot{\mathtt{L}}$ thus satisfies Klein identities also. In general, lines of $\dot{\mathtt{L}}$ are linearly independent. Therefore, there is exactly one $\dot{\mathtt{Q}}$ of coefficients that linearly combines the rows of $\dot{\mathtt{L}}$ into $\mathtt{S}$ as $\mathtt{S} = \dot{\mathtt{Q}}\dot{\mathtt{L}}$. Matrices $\hat{\mathtt{Q}}_{9\times 6}$, $\dot{\mathtt{Q}}_{9\times 6}$ are in general of rank six and thus (7.5) fixes $\mathtt{H}$ uniquely. Since there is exactly one $\dot{\mathtt{Q}}$ so that $\mathtt{S} = \dot{\mathtt{Q}}\dot{\mathtt{L}}$ the linear mapping bringing $\hat{\mathtt{Q}}$ into $\mathtt{Q}$ (which exists by the argument given in the next paragraph) equals $\mathtt{H}$. □

When there is noise in the data, $\mathtt{H}$ obtained according to equation (7.5) does not have to fulfill Klein identities (7.4). Nonlinear bundle adjustment with initial solution $\mathtt{H}$ should be used. Another, but only approximate, solution is obtained by finding some linear projections of each

Scene *Cubes*	5 images [576 × 768], 14 correspondences, manual detection		
Method \ γ_l^i estimation	*sequence, two view overlap*	*two central images*	*no factorization*
Factorization	8.69 / 155.14 / 2.46	6.03 / 74.24 / 2.81	
Linear method	**3.80** / 24.03 / 2.45	**2.42** / 13.41 / 1.66	**1.89** / 11.09 / 1.08
Linear method + BA	**0.47** / 2.12 / 0.34	**0.47** / 2.12 / 0.33	**0.47** / 2.12 / 0.33



Scene *House (Oxford)*	6 images [576 × 768], 31 correspondences, automatic detection		
Method \ γ_l^i estimation	*sequence, two view overlap*	*two central images*	*no factorization*
Factorization	4.62 / 42.98 / 2.40	3.38 / 30.17 / 1.91	
Linear method	**1.57** / 23.24 / 0.77	**0.80** / 10.53 / 0.44	**1.03** / 13.33 / 0.52
Linear method + BA	**0.23** / 1.32 / 0.17	**0.23** / 1.32 / 0.17	**0.23** / 1.32 / 0.17

Figure 7.1: *Experiments with real scenes.* Mean / maximum / median reprojection errors are shown.

row of $\hat{\mathtt{Q}}$, resp. column of $\hat{\mathtt{L}}$, onto the Klein quadric. If there is noise in the data, matrix $\mathtt{H}$ from proposition 7 does not have to exist. However, it is always possible to find $\mathtt{H}$ using a partial reconstruction from three views so that equation (7.5) holds. It turned out in our experiments that although rows of $\hat{\mathtt{Q}}\mathtt{H}$, resp. columns of $\mathtt{H}^{-1}\hat{\mathtt{L}}$, do no satisfy the Klein identities, they are close to the Klein quadric so that a good solution can be obtained by projecting them onto the Klein quadric.

Projection onto the Klein quadric was done in the following way. For each view, i, system of linear equations

$$\mathbf{l}_l^{i\top}\mathtt{P}^i\tilde{\mathbf{X}}_{l,p} = 0 \quad \text{for} \quad l = 1\dots n, \quad p = 1,2 \tag{7.8}$$

was used to estimate point projection matrix $\mathtt{P}^i$ from image measurements and matrix $\tilde{\mathtt{L}} = \mathtt{H}^{-1}\hat{\mathtt{L}}$ where $\tilde{\mathbf{X}}_{l,p} \in \mathbb{R}^4$ are some two columns of the dual Plücker matrix of $\tilde{\mathbf{L}}_l$. Finally, each line on the Klein quadric was estimated by intersecting backprojections of the image measurements using all point projection matrices as in step 4 of algorithm 7. Our method for scene reconstruction from lines is summarized in algorithm 8.

Implementation Details

On account of good numerical conditioning, several normalizations of the data and balancing similar to those in [113] need to be performed.

Experiments

The method has been tested on a simulated scene and on two real scenes. No point correspondences have been used for the line reconstruction by algorithm 8. The reconstructed 3-space lines were reprojected into the images. The reprojection error of a line was computed as the mean of Euclidean distances between the end-points of the original image line and the reprojected reconstructed line.

An artificial scene was used for an experiment with simulated data. 20 images of 30 lines in space have been obtained from different viewpoints. The reconstruction was precise in absence of noise and the mean error of the reconstruction grew linearly with the variance of the added noise.

The tables describing each real experiment in figure 7.1 include scene name, number of images and their sizes, number of correspondences together with the way of their detection. The reconstruction method was used in two setups of image triplets: (i) sequence with two view overlap and (ii) two central images. For each setup, reprojection errors of the plain factorization into $\hat{\mathtt{Q}}\hat{\mathtt{L}}$, of the linear method, and of the non-linear line bundle adjustment initiated by the linear method are shown. For comparison, the following simple reconstruction method without

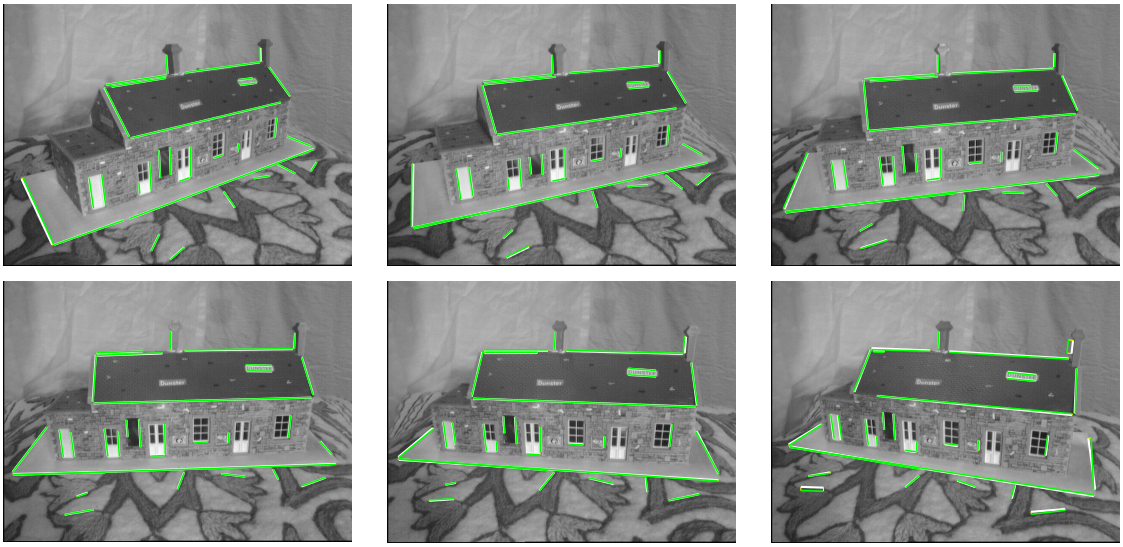


Figure 7.2: *Line reconstruction from many images.* No point correspondences have been used for the line reconstruction by algorithm 8. Mean / maximum reprojection errors of the method without bundle adjustment were 0.80 / 10.53 pxl.

factorization is given. (i) L from a partial reconstruction of an image triplet was used to estimate point projection matrices using equation (7.8). Then, (ii) backprojections from all images were intersected as described in section 7.1.

In the Cube scene, five images of cubes on a checkerboard have been obtained from different viewpoints, three of them can be seen in figure 1.2, 5. 14 correspondences of lines, at least partially visible in all images, have been detected manually. Mean, resp. maximum, reprojection errors of the linear method of the better setup were 2.42, resp. 13.41, pixels. The House scene was obtained on six images, see figure 7.2. The original lines are drawn in white and the reprojected ones in green color. Mean (resp., maximum) reprojection errors of the linear method of the better setup were 0.80 (resp., 10.53), pixels.

In both experiments with real scenes, the way of scale factor estimation with two central images provided a better solution. Both factorizations of the Cube scene provided worse solutions than the reconstruction without factorization. This may be due to a little amount (14) of correspondences and unprecise manual line detection. On the other hand, 31 automatic correspondences in the House scene enabled improvement of the factorization by 20% compared to the simple method.

Summary and Conclusions

A linear method for line reconstruction via a factorization of a line measurement matrix has been presented and tested on simulated and real scenes. The method is, in principle, capable to reconstruct lines even though there are no corresponding points on the lines available. We have used line projection by a perspective camera to formulate the reconstruction as a factorization and showed how to carry it out in a noise free situation.

We have pointed out that finding the optimal reconstruction w.r.t. noise in data is, as usual, a non-linear task but demonstrated that the vectors obtained by plain SVD factorization followed by a transformation considering the constraint (2.6) provide a good approximate solution that may be hoped as a starting point for nonlinear bundle adjustment.

The line factorization method can be straightforwardly extended to deal with occluded lines using Jacobs' algorithm [43] as it was performed in [62]. There exist many ways how to establish triplets of views in step 1 of algorithm 8. In presence of noise, reconstructions resulting from different sets of triplets differ in the reprojection error. Similar heuristics for choosing the best set of triplets based on the structure of the missing data to those in [62] can be used.

7.2 Metric Gluing

Projective reconstructions of camera triplets were obtained in [64] (section 7.1) using the trifocal tensor. It is easy to make metric upgrades of each camera triplet using the technique for upgrading a fundamental matrix (obtained from each camera pair) into an essential matrix described in section 4.2. Possible inconsistencies among the three pair-wise metric reconstructions can be solved, e.g., using bundle adjustment.

As noted in the example for points (chapter 6, *p*41), the metric three-view reconstructions can be easily glued projectively (sections 6.3 or 6.5.3) or by first registering rotations and then translations (sections 6.7 or 6.8). Note that all these methods naturally handle occlusions for lines as well.

This method has not been implemented as our interests were focused on other things and lines are not very useful features in general except for man-made environments.

8 Conclusions

This thesis provided several methods for 3D multiview reconstruction. Various features were considered (points and lines) as well as camera models (affine, perspective, and omnidirectional) and types of reconstruction (projective, oriented-projective, and metric). The newest methods are highly robust w.r.t. occlusions (99.9% of the data is missing), mismatches and even non-existent epipolar geometries.

While studying the new multiview algorithms, new insights into the substance of difficult problems were needed where the traditional methods like factorization or bundle adjustment did not work. Examples include "ignoring" the missing data problem by looking at the known data only [65] (section 6.3) and fitting a Gaussian to the inliers of an epipolar geometry and removing the furthest points in the mismatch removal problem [67] (section 6.8.3).

As a side effect of processing very sparse and large data, many other problems had to be solved. These contained robust two- and three-view geometry estimation for uncalibrated, partially-, and fully-calibrated cameras which is capable of working under extreme conditions such as small image overlap, m-n tentative correspondences and presence of a dominant plane. A huge data had to be handled from both memory and speed points of view. Heuristics for reducing the number of the matched image pairs were given. Integration with dense stereo and fish-scales generation was continuously improved as more difficult scenes were processed. As a result, a fully automatical pipeline starting from images and ending at 3D vrml models was built.

The images of the scenes presented in the newest paper [67] (section 6.8) were made public on the demo page [3]. Sofar, we do not know about any other method that would be able to reconstruct any of the scenes, showing so outstanding capabilities of the developed method.

The pipeline was used in the ICCV05 Vision Contest [59] (section 6.6) and is continuously used in the Center for Machine Perception in Prague for obtaining reconstructions exploited for instance as a ground truth for comparison with incremental methods. Recently most of the developed code was sold to a software company, demonstrating so its practicality and reliability to work well under very distinct conditions.

A Best Rank-one Approximation

In this chapter, an attempt is made to find a best rank-one approximation to a matrix with missing elements. It was motivated by the following idea. If a good rank-one approximation is found to a matrix with missing elements, one could subtract if from the matrix and so decrease its rank by one and repeat this several times. For instance, it would be repeated four-times on the rescaled measurement matrix to obtain a projective reconstruction by factorization, see equation (2.2), p7.

Unfortunately, the rank-one approximation proposed here turned out to be not satisfactory for obtaining reasonable results in 3D reconstruction, showing so how difficult problem it is. Note that practical solutions to the missing data problem were given by combining (gluing) partial factorizations (reconstructions) in [65, 67] (sections 6.3, 6.8).

Factorization of a matrix with missing elements, $\mathtt{A} \in \mathbb{R}^{m \times n}$, into a product of rank-one matrices $\mathbf{b} \in \mathbb{R}^{m \times 1}$ and $\mathbf{c}^\top \in \mathbb{R}^{1 \times n}$ is searched for:

$$\underbrace{\begin{bmatrix} a_1^1 & a_2^1 & \dots & a_n^1 \\ \times & a_2^2 & \dots & \times \\ \vdots & & \ddots & \vdots \\ a_1^m & \times & \dots & a_n^m \end{bmatrix}}_{\mathtt{A}} = \underbrace{\begin{pmatrix} b^1 \\ \vdots \\ b^m \end{pmatrix}}_{\substack{\mathbf{b} \\ m \times 1}} \underbrace{[c_1 \dots c_n]}_{\substack{\mathbf{c}^\top \\ 1 \times n}} \tag{A.1}$$

where marks $\times$ stand for unknown elements.

For each $p \in \{1, \dots, n\}$, let $\mathbf{i}_p = \{i_p^1, \dots, i_p^{|\mathbf{i}_p|}\}$ denote indices of all rows for which a_p^i, $i \in \mathbf{i}_p$, are known. The following equation with no missing elements holds:

$$\begin{pmatrix} a_p^{i_p^1} \\ \vdots \\ a_p^{i_p^{|\mathbf{i}_p|}} \end{pmatrix} = \begin{pmatrix} b^{i_p^1} \\ \vdots \\ b^{i_p^{|\mathbf{i}_p|}} \end{pmatrix} c_p.$$

Let $\mathbf{a}_p^{\mathbf{i}_p}$ denote $\begin{pmatrix} a_p^{i_p^1} \\ \vdots \\ a_p^{i_p^{|\mathbf{i}_p|}} \end{pmatrix}$ and let $\mathbf{b}^{\mathbf{i}_p}$ denote $\begin{pmatrix} b^{i_p^1} \\ \vdots \\ b^{i_p^{|\mathbf{i}_p|}} \end{pmatrix}$. Equation $\mathbf{a}_p^{\mathbf{i}_p} = \mathbf{b}^{\mathbf{i}_p} c_p$ is bilinear in both unknowns $\mathbf{b}^{\mathbf{i}_p}$ and c_p. It can be linearized by multiplying both sides of the equation by the inverse of c_p:

$$\mathbf{a}_p^{\mathbf{i}_p} \frac{1}{c_p} = \mathbf{b}^{\mathbf{i}_p}.$$

Let h_p denote $\frac{1}{c_p}$, $h_p \in \mathbb{R}$. Let the following system of equation is solved:

$$\mathbf{a}_p^{\mathbf{i}_p} h_p = \mathbf{b}^{\mathbf{i}_p} \quad \text{for } p = 1, \dots, n \tag{A.2}$$

What Is Minimized

Solution to (A.1) in the least square sense minimizes the Frobenius norm (for vector $\mathbf{a} \in \mathbb{R}^n$, $\|\mathbf{a}\| = \sqrt{\sum_{i=1}^{n} a_i{}^2}$) between the pth data column and the pth approximation column, er_{Frob}:

$$er_{Frob} = \left\| \mathbf{a}_p^{\mathbf{i}_p} - \mathbf{b}^{\mathbf{i}_p} c_p \right\|.$$

In equations (A.2), error

$$\begin{aligned} er_{min} &= \left\| \mathbf{a}_p^{\mathbf{i}_p} \frac{1}{c_p} - \mathbf{b}^{\mathbf{i}_p} \right\| \\ &= \left\| \frac{1}{c_p} \left(\mathbf{a}_p^{\mathbf{i}_p} - \mathbf{b}^{\mathbf{i}_p} c_p \right) \right\| \\ &= \frac{1}{|c_p|} \left\| \mathbf{a}_p^{\mathbf{i}_p} - \mathbf{b}^{\mathbf{i}_p} c_p \right\| \\ &= \frac{1}{|c_p|} er_{Frob} \end{aligned}$$

is minimized. This means, in column p, er_{min} differs from er_{Frob} by $\frac{1}{|c_p|}$. This means the solution is not optimal even in the case of complete data. It turned out that it is possible to estimate both $\mathbf{b}$ and $\mathbf{c}^\top$ so that exactly er_{Frob} is minimized for complete data and a good approximation of er_{Frob} is minimized for incomplete data.

Rank-one approximation of $\mathtt{A}$ such that $\mathbf{a}_p^{\mathbf{i}_p} = \mathbf{b}^{\mathbf{i}_p} c_p$ is searched for. However, this expression is bilinear. When is it equivalent to $\mathbf{a}_p^{\mathbf{i}_p} c_p = \mathbf{b}^{\mathbf{i}_p}$ in the least squares sense? That is when $\left\| \mathbf{a}_p^{\mathbf{i}_p} - \mathbf{b}^{\mathbf{i}_p} c_p \right\| = \left\| \mathbf{a}_p^{\mathbf{i}_p} c_p - \mathbf{b}^{\mathbf{i}_p} \right\|$? The following theorem holds:

Theorem 2 (Length Swap Invariant) *For any vectors* $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ *and any scalar* c, $c \in \mathbb{R}$*:*

$$\|\mathbf{a} - \mathbf{b}c\| = \|\mathbf{a}c - \mathbf{b}\| \text{ if and only if } \|\mathbf{a}\| = \|\mathbf{b}\|.$$

Proof. $\mathbf{a} = (a_1, \ldots, a_n)^\top$, $\mathbf{b} = (b_1, \ldots, b_n)^\top$. The following equations are equivalent:

$$\begin{aligned} \sqrt{\sum_i (a_i - b_i c)^2} &= \sqrt{\sum_i (a_i c - b_i)^2} \\ \sum (a_i - b_i c)^2 &= \sum (a_i c - b_i)^2 \\ \sum a_i^2 - 2c \sum a_i b_i + c^2 \sum b_i^2 &= c^2 \sum a_i^2 - 2c \sum a_i b_i + \sum b_i^2 \\ \|\mathbf{a}\|^2 + c^2 \|\mathbf{b}\|^2 &= c^2 \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 \\ \|\mathbf{a}\|^2 (1 - c^2) &= \|\mathbf{b}\|^2 (1 - c^2) \\ \|\mathbf{a}\| &= \|\mathbf{b}\| \end{aligned}$$

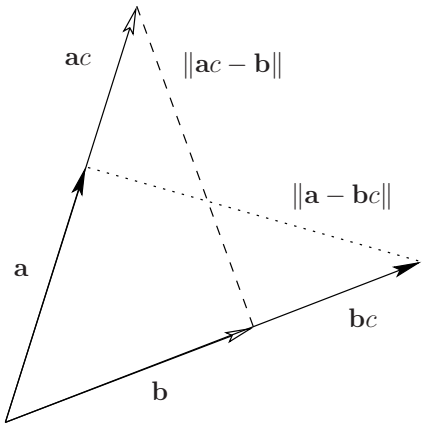
□

Theorem 2 says that norm of the difference between two vectors stays same after swapping their lengths, see figure A.1.

Theorem 3 (Rank-one Approximation for Complete Data) *In case of complete data, the least square solution to the system of equations*

$$\left(\mathbf{a}_p \mathbf{a}_p^\top \frac{1}{\|\mathbf{a}_p\|} - \|\mathbf{a}_p\| \right) \mathbf{b} = 0 \quad \text{for } p = 1, \ldots, n$$

gives the best rank-one approximation to matrix $\mathtt{A}$ *in Frobenius norm as* $\mathtt{A} = \mathbf{b}\mathbf{c}^\top$ *where* $c_p = \mathbf{b}^\top \mathbf{a}_p$.



$$\|\mathbf{a} - \mathbf{b}c\| = \|\mathbf{a}c - \mathbf{b}\| \text{ iff } \|\mathbf{a}\| = \|\mathbf{b}\|$$

Figure A.1: Length Swap Invariant: norm of the difference between two vectors stays same after swapping their lengths

Note to theorem 3. $\mathbf{b}$ and $\frac{\mathbf{c}^\top}{\|\mathbf{c}\|}$ are the most significant column and row singular vectors, respectively. $\|\mathbf{c}\| = \|\mathtt{A}\|$ is the most significant singular number of $\mathtt{A}$.

Proof of theorem 3. Using theorem 2,

$$\begin{aligned}\|\mathbf{a}_p - \mathbf{b}c_p\| &= \left\| \underbrace{\mathbf{a}_p}_{\text{length } \|\mathbf{a}_p\|} - \underbrace{\mathbf{b}\frac{\|\mathbf{a}_p\|}{\|\mathbf{b}\|}}_{\text{length } \|\mathbf{a}_p\|} \underbrace{\frac{\|\mathbf{b}\|}{\|\mathbf{a}_p\|}c_p}_{c} \right\| = \\ &= \left\| \mathbf{a}_p \underbrace{\frac{\|\mathbf{b}\|}{\|\mathbf{a}_p\|}c_p}_{c} - \mathbf{b}\frac{\|\mathbf{a}_p\|}{\|\mathbf{b}\|} \right\| = \dots\end{aligned}$$

After substituting for c_p by its least square estimate[1], $c_p = \frac{\mathbf{b}^\top \mathbf{a}_p}{\|\mathbf{b}\|^2}$, one gets

$$\begin{aligned}\dots &= \left\| \mathbf{a}_p \mathbf{b}^\top \mathbf{a}_p \frac{1}{\|\mathbf{a}_p\|\|\mathbf{b}\|} - \mathbf{b}\frac{\|\mathbf{a}_p\|}{\|\mathbf{b}\|} \right\| = \\ &= \left\| \left(\mathbf{a}_p \mathbf{a}_p^\top \frac{1}{\|\mathbf{a}_p\|} - \|\mathbf{a}_p\| \right) \frac{\mathbf{b}}{\|\mathbf{b}\|} \right\|.\end{aligned}$$

For solutions of systems of equations are searched for as unit vectors, $\|\mathbf{b}\|$ can be substituted for 1. □

$\|\mathbf{a}_p\|$ expresses the mass of the pth column. In case of incomplete data,

$$\left\| \mathbf{a}_p^{\mathbf{i}_p} - \mathbf{b}^{\mathbf{i}_p} c_p \right\| = \left\| \left(\mathbf{a}_p^{\mathbf{i}_p} \mathbf{a}_p^{\mathbf{i}_p \top} \frac{1}{\|\mathbf{a}_p^{\mathbf{i}_p}\|} - \|\mathbf{a}_p^{\mathbf{i}_p}\| \right) \frac{\mathbf{b}^{\mathbf{i}_p}}{\|\mathbf{b}^{\mathbf{i}_p}\|} \right\|.$$

[1] similar to the depth estimation [113]

Here, $\|\mathbf{b}^{\mathbf{i}_p}\|$ is unknown. It can be well approximated by the mass in rows $\mathbf{i}_p$:

$$\|\mathbf{b}^{\mathbf{i}_p}\| = \frac{\sqrt{\sum_{i\in\mathbf{i}_p}\sum_{p,\ a_p^i \text{ known}} {a_p^i}^2}}{\sqrt{\sum_{i=1}^{n}\sum_{p,\ a_p^i \text{ known}} {a_p^i}^2}}.$$

This approximation can be used as an initialization for an iterative process of improving the estimate of $\mathbf{b}$ using the actual approximation of $\|\mathbf{b}^{\mathbf{i}_p}\|$, $p = 1, \ldots, n$.

It turned out in our experiments that sometimes the error of the solution gets worse. A possible reason is that it is far from the (global) minimum. Several steps of the gradient descent on $\mathbf{b}$ could be tried when this happens. It is possible that this equation gives a "longer" step than the gradient descent if it improves. However, this has not been studied in detail yet.

Bibliography

[1] http://cmp.felk.cvut.cz/~martid1/demoCVPR05. 6, 68

[2] http://cmp.felk.cvut.cz/~martid1/demo3DPVT06. 6, 37, 96, 97

[3] http://cmp.felk.cvut.cz/~martid1/demoCVPR07. 6, 109, 117

[4] http://research.microsoft.com/iccv2005/Contest. 82, 95

[5] H. Aanæs, R. Fisker, K. Åström, and J. M. Carstensen. Robust factorization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(9):1215–1225, 2002. 11

[6] A. Akbarzadeh, J.-M. Frahm, P. Mordohai, B. Clipp, C. Engels, D. Gallup, P. Merrell, M. Phelps, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewenius, R. Yang, G. Welch, H. Towles, D. Nister, and M. Pollefeys. Towards urban 3D reconstruction from video. In *3DPVT*, University of North Carolina, Chapel Hill, USA, June 2006. CD-ROM. 11

[7] S. Avidan and A. Shashua. Threading fundamental matrices. In *IEEE Trans. on PAMI*, vol. 23(1), pp. 73–77, 2001. 4, 9, 10, 43

[8] M. K. Bennett. *Affine and Projective Geometry*. John Willey and Sons, New York, USA, 1995. 44

[9] S. Bougnoux. From projective to euclidean space under any practical situation, a criticism of self-calibration. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, p. 790, 1998. 21

[10] M. Brand. Incremental singular value decomposition of uncertain data with missing values. In *ECCV*, 2002. 54, 103

[11] A. M. Buchanan and A. W. Fitzgibbon. Damped newton algorithms for matrix factorization with missing data. In *CVPR05*, vol. 2, pp. 316–322, 2005. 70, 71

[12] J. Čech and R. Šára. Efficient sampling of disparity space for fast and accurate matching. In *Proc. BenCOS Workshop CVPR*, 2007. To appear. 14, 29, 81, 98, 105, 106

[13] O. Chum, J. Matas, and J. Kittler. Locally optimized RANSAC. In *DAGM*, pp. 236–243, 2003. 19, 22, 23, 24, 36, 37

[14] O. Chum, J. Matas, and Š. Obdržálek. Enhancing RANSAC by generalized model optimization. In *Proc. of the Asian Conference on Computer Vision (ACCV)*, vol. 2, pp. 812–817, Jeju Island, Korea South, January 2004. 22

[15] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Proc. ICCV*, 2007. 31

[16] O. Chum, T. Werner, and J. Matas. Two-view geometry estimation unaffected by a dominant plane. In *CVPR*, vol. 1, pp. 772–779, 2005. 5, 6, 12, 14, 22, 24, 37, 82, 84, 86, 88, 108

[17] H. Cornelius, R. Šára, D. Martinec, T. Pajdla, O. Chum, and J. Matas. Towards complete free-form reconstruction of complex 3D scenes from an unordered set of uncalibrated images. In *SMVP/ECCV*, vol. LNCS 3247, pp. 1–12, Prague, Czech Republic, May 2004. 14, 36, 41, 42, 61, 67, 68, 95, 96, 105, 106

[18] C. Engels, H. Stewénius, and D. Nistér. Bundle adjustment rules. In *Photogrammetric Computer Vision (PCV)*, Sept. 2006. 36

[19] D. Fidaleo, G. G. Medioni, P. Fua, and V. Lepetit. An investigation of model bias in 3d face tracking. In *ICCV Workshop on Analysis and Modelling of Faces and Gestures*, pp. 125–139, Beijing, China, October 2005. 81

[20] A. Fitzgibbon. Robust registration of 2d and 3d point sets. In *Proceedings of the British Machine Vision Conference*, vol. II, pp. 411–420, Manchester, UK, September 2001. 36

[21] A. W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *Proc. ECCV*, vol. I, pp. 311–326, June 1998. 4, 9, 10, 43

[22] D. Forsyth, S. Ioffe, and J. Haddon. Bayesian structure from motion. In *ICCV*, pp. 660–665, 1999. 91

[23] J.-M. Frahm and M. Pollefeys. Ransac for (quasi-)degenereate data (qdegsac). In *Proc. CVPR*, 2006. 12

[24] D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys. Real-time plane-sweeping stereo with multiple sweeping directions. In *Proc. CVPR*, 2007. 17

[25] J. Goldberger. Reconstructing camera projection matrices from multiple pairwise overlapping views. *Comput. Vis. Image Underst.*, 97(3):283–296, 2005. 77
[26] V. Govindu. Combining two-view constraints for motion estimation. In *Proc. CVPR*, 2001. 100
[27] N. Guilbert and A. Bartoli. Batch recovery of multiple views with missing data using direct sparse solvers. In *Proceedings of the British Machine Vision Conference*, 2003. 4, 9, 10, 54, 61, 67, 68, 70
[28] C. J. Harris and M. Stephens. A combined corner and edge detector. In *Proc. of Alvey Vision Conference*, pp. 147–151, 1988. 50
[29] R. Hartley and F. Kahl. Global optimization through searching rotation space and optimal estimation of the essential matrix. In *Proc. ICCV*, 2007. 11
[30] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2000. 7, 8, 9, 12, 43, 48, 112
[31] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University, Cambridge, 2nd edition, 2003. 3, 6, 9, 11, 12, 13, 20, 21, 22, 25, 30, 35, 37, 39, 40, 48, 58, 61, 63, 74, 77, 79, 80, 86, 87, 90, 103
[32] R. I. Hartley. Theory and practice of projective rectification. *IJCV*, 35(2):115–127, November 1999. 14, 29
[33] R. I. Hartley, E. Hayman, L. d. Agapito, and I. D. Reid. Camera calibration and the search for infinity. In *Proc. ICCV*, vol. 1, pp. 510–517, 1999. 10, 68, 82, 86
[34] R. I. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding: CVIU*, 68(2):146–157, 1997. 12
[35] A. Heyden. Projective structure and motion from image sequences using subspace methods. In *Proc. 10th SCIA*, pp. 963–968, June 1997. 9, 10, 43
[36] B. K. P. Horn. Closed form solution of absolute orientation using unit quaternions. *Journal of the Optical Society A*, 4(4):629–642, April 1987. 99
[37] C.-R. Huang, C.-S. Chen, and P.-C. Chung. An improved algorithm for two-image camera self-calibration and euclidean structure recovery using absolute quadric. *Pattern Recognition*, 37(8):1713–1722, 2004. 20, 21
[38] D. Q. Huynh and A. Heyden. Outlier detection in video sequences under affine projection. In *Proc. IEEE Conf. on CVPR*, vol. I, pp. 695–701, Kauai, Hawaii, 9-14, December 2001. 10
[39] D. Q. Huynh and A. Heyden. Outlier detection in video sequences under affine projection. In *Proc. of CVPR*, vol. I, pp. 695–701, 2001. 52
[40] A. Ibrahimbegovic. On the choice of finite rotation parameters. *Computer Methods in Applied Mechanics and Engineering*, 149(1):49–71, October 1997. 35
[41] M. Irani and P. Anandan. Parallax geometry of pairs of points for 3D scene analysis. In *ECCV (1)*, pp. 17–30, 1996. 22
[42] M. Irani and P. Anandan. Factorization with uncertainty. In *ECCV (1)*, pp. 539–553, 2000. 11
[43] D. Jacobs. Linear fitting with missing data: Applications to structure from motion and to characterizing intensity images. In *CVPR*, pp. 206–212, 1997. 4, 5, 9, 10, 39, 42, 43, 44, 46, 47, 52, 54, 56, 57, 58, 71, 116
[44] F. Kahl. Multiple view geometry and the L_∞-norm. In *ICCV05*, pp. II: 1002–1009, 2005. 6, 11, 24, 25, 26, 27, 30, 35, 40, 42, 60, 82, 83, 86, 90, 98, 102, 104, 105, 106
[45] F. Kahl and A. Heyden. Affine structure and motion from points, lines and conics. In *IJCV 33(3)*, pp. 163–180, 1999. 12
[46] G. Kamberov, G. Kamberova, O. Chum, Š. Obdržálek, D. Martinec, J. Kostková, T. Pajdla, J. Matas, and R. Šára. 3D geometry from uncalibrated images. In *ISVC '06: Proceedings 2nd International Symposium on Visual Computing*, number 4292 in Lecture Notes in Computer Science, pp. 802–813, Lake Tahoe, USA, November 2006. 14, 16
[47] T. Kanade and D. Morris. Factorization methods for structure from motion. *Phil. Trans. R. Soc. Lond. A*, 356(1740):1153–1173, 1998. 12
[48] K. Kanatani. Model selection in statistical inference and geometric fitting. In *Proc. 3rd Workshop on Information-Based Induction Sciences*, Izu, Japan, July 2000. 12, 84
[49] Q. Ke and T. Kanade. Quasiconvex optimization for robust geometric reconstruction. In *ICCV*, pp. 986–993, 2005. 11, 25
[50] R. Koch, M. Pollefeys, and L. J. V. Gool. Robust calibration and 3d geometric modeling from large collections of uncalibrated images. In *DAGM-Symposium*, pp. 413–420, 1999. 10
[51] R. Koch, M. Pollefeys, and L. Van Gool. Multi viewpoint stereo from uncalibrated video sequences. In *ECCV*, vol. 1, pp. 55–71, 1998. 11, 16

[52] J. Kostková and R. Šára. Stratified dense matching for stereopsis in complex scenes. In *BMVC 2003: Proceedings of the 14th British Machine Vision Conference*, vol. 1, pp. 339–348, Norwich, UK, September 2003. 14, 16, 29, 42, 68, 81

[53] N. Levi and M. Werman. The viewing graph. In *CVPR*, vol. I, pp. 518–524, 2003. 11

[54] M. Lourakis and A. Argyros. The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm. Technical Report 340, Institute of Computer Science - FORTH, Heraklion, Crete, Greece, Aug 2004. 35, 36, 85, 86, 98

[55] M. Lourakis and A. Argyros. Fast trifocal tensor estimation using virtual parallax. In *Proceedings of the IEEE International Conference on Image Processing (ICIP 2005)*, pp. 169–172, Genoa, Italy, June 2005. 25

[56] D. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, vol. 20, pp. 91–110, 2003. 31, 33, 108

[57] Q.-T. Luong and O. D. Faugeras. A stability analysis of the fundamental matrix. In *ECCV (1)*, pp. 577–588, 1994. 91

[58] S. Mahamud and M. Hebert. Iterative projective reconstruction from multiple views. In *CVPR*, 2000. 9, 10, 43

[59] D. Martinec, J. Matas, M. Perďoch, O. Chum, Š. Obdržálek, T. Pajdla, and T. Werner. The second place in iccv2005 computer vision contest. http://cmp.felk.cvut.cz/publicity/iccv-contest05, October 2005. 5, 42, 83, 117

[60] D. Martinec and T. Pajdla. Structure from many perspective images with occlusions. Research Report CTU–CMP–2001–20, Center for Machine Perception, K333 FEE Czech Technical University, Prague, Czech Republic, July 2001. 46, 48, 49, 52

[61] D. Martinec and T. Pajdla. Outlier detection for factorization-based reconstruction from perspective images with occlusions. In *Proceedings of the Photogrammetric Computer Vision (PCV)*, vol. B, pp. 161–164, Graz, Austria, September 2002. 6, 33, 39, 42, 53

[62] D. Martinec and T. Pajdla. Structure from many perspective images with occlusions. In *Proc. of the European Conference on Computer Vision (ECCV)*, vol. II, pp. 355–369, Copenhagen, Denmark, May 2002. 6, 35, 39, 42, 43, 52, 53, 116

[63] D. Martinec and T. Pajdla. Consistent multi-view reconstruction from epipolar geometries with outliers. In *Proceedings of the 13th Scandinavian Conference on Image Analysis (SCIA)*, pp. 493–500, Göteborg, Sweden, June 2003. 4, 6, 33, 42

[64] D. Martinec and T. Pajdla. Line reconstruction from many perspective images by factorization. In *Proceedings of the Computer Vision and Pattern Recognition conference (CVPR)*, vol. I, pp. 497–502, Madison, Wisconsin, USA, June 2003. 6, 110, 112, 116

[65] D. Martinec and T. Pajdla. 3D reconstruction by fitting low-rank matrices with missing data. In *Proc CVPR*, vol. I, pp. 198–205, San Diego, CA, USA, June 2005. 4, 5, 10, 23, 24, 34, 39, 40, 42, 44, 51, 52, 54, 70, 71, 72, 77, 78, 79, 81, 82, 83, 86, 87, 88, 89, 96, 100, 117, 118

[66] D. Martinec and T. Pajdla. 3D reconstruction by gluing pair-wise Euclidean reconstructions, or "how to achieve a good reconstruction from bad images". In *3DPVT*, University of North Carolina, Chapel Hill, USA, June 2006. CD-ROM. 5, 14, 22, 23, 24, 32, 34, 40, 42, 60, 85, 86, 88, 96, 97, 98, 100, 104

[67] D. Martinec and T. Pajdla. Robust rotation and translation estimation. In *Proc CVPR*, Minneapolis, Minnesota, USA, June 2007. CD-ROM. 5, 20, 25, 27, 30, 33, 34, 36, 37, 39, 40, 42, 98, 109, 117, 118

[68] J. Matas and O. Chum. Randomized ransac with sequential probability ratio test. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, vol. II, pp. 1727–1732, Hotel Beijing, Beijing, China, October 2005. 22

[69] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, vol. 1, pp. 384–393, Cardiff, UK, September 2002. 6, 14, 22, 42, 53, 79

[70] J. Matas, Š. Obdržálek, and O. Chum. Local affine frames for wide-baseline stereo. In *ICPR*, vol. 4, pp. 363–366, 2002. 14, 29, 33, 95, 108

[71] M. Matoušek. *Epipolar Rectification Minimising Image Loss*. Phd thesis, Center for Machine Perception, K13133 FEE Czech Technical University, Prague, Czech Republic, October 2007. 14, 29

[72] M. Matoušek, R. Šára, and V. Hlaváč. Data-optimal rectification for fast and accurate stereovision. In *Proceedings of the Third International Conference on Image and Graphics*, pp. 212–215, Hong Kong, China, December 2004. 14

[73] B. Mičušík, D. Martinec, and T. Pajdla. 3D metric reconstruction from uncalibrated omnidirectional images. In *Proc. of the Asian Conference on Computer Vision (ACCV)*, vol. 1, pp. 545–550, Jeju Island, Korea, January 2004. 6, 40, 42, 54, 68

[74] B. Mičušík and T. Pajdla. Estimation of omnidirectional camera model from epipolar geometry. In *CVPR*, vol. I, pp. 485–490, 2003. 6, 42

[75] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A Comparison of Affine Region Detectors. *IJCV*, 2005. 14, 33, 95, 108

[76] D. Nistér. *Automatic dense reconstruction from uncalibrated video sequences.* PhD thesis, Royal Institute of Technology KTH, Stockholm, Sweden, March 2001. 10, 35

[77] D. Nistér. An efficient solution to the five-point relative pose. *PAMI*, 26(6):756–770, June 2004. 11, 20, 22, 82, 88, 98, 108

[78] D. Nistér. Untwisting a projective reconstruction. *IJCV*, 60(2):165–183, November 2004. 10, 68, 79, 81, 86

[79] D. Nistér, F. Kahl, and H. Stewénius. Structure from motion with missing data is np-hard. In *Proc. ICCV*, 2007. 3

[80] Š. Obdržálek and J. Matas. Sub-linear indexing for large scale object recognition. In *BMVC 2005: Proceedings of the 16th British Machine Vision Conference*, vol. 1, pp. 1–10, Oxford, UK, September 2005. 31

[81] J. Oliensis. Exact two–image structure from motion. *PAMI*, 24(12):1618–1633, 2002. 12

[82] C. Olsson, A. P. Eriksson, and F. Kahl. Efficient optimization for l-infinity problems using pseudoconvexity. In *Proc. ICCV*, 2007. 11

[83] C. Olsson, F. Kahl, and M. Oskarsson. Optimal estimation of perspective camera pose. In *Proc. International Conference on Pattern Recognition*, 2006. 11

[84] C. Olsson, F. Kahl, and M. Oskarsson. The registration problem revisited: Optimal solutions from points, lines and planes. In *Proc. Computer Vision and Pattern Recognition*, 2006. 11

[85] M. Oskarsson, A. Zisserman, and K. Astrom. Minimal projective reconstruction for combinations of points and lines in three views. In *Proceedings of the British Machine Vision Conference*, 2002. 12

[86] M. Perďoch, J. Matas, and O. Chum. Epipolar geometry from two correspondences. In *ICPR 2006: Proceedings of the 18th International Conference on Pattern Recognition*, vol. 4, pp. 215–220, Hong Kong, China, August 2006. 22

[87] J. Philbin, O. Chum, M. Isard, J. Sivic, , and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proc. CVPR*, 2007. 31

[88] J. Philip. A non-iterative algorithm for determining all essential matrices corresponding to five point pairs. *Photogrammetric Record*, 15(88):589–599, October 1996. 21

[89] J. Philip. Critical point configurations of the 5-, 6-, 7- and 8-point algorithms for relative orientation. Technical report, Department of Mathematics, Royal Institute of Technology, Stockholm, 1998. 21

[90] M. Pollefeys, L. Gool, M. Vergauwen, K. Cornelis, F. Verbiest, and J. Tops. Video-to-3d. In *Proceedings of Photogrammetric Computer Vision 2002 (ISPRS Commission III Symposium), International Archive of Photogrammetry and Remote Sensing*, vol. 34, pp. 252–258, 2002. 12, 84

[91] M. Pollefeys, L. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *Int. J. Comput. Vision*, 59(3):207–232, 2004. 10

[92] M. Pollefeys, R. Koch, and L. V. Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Proc. International Conference on Computer Vision*, pp. 90–95, Bombay, 1998. 10

[93] M. Pollefeys, R. Koch, and L. Van Gool. A simple and efficient rectification method for general motion. In *Proc. International Conference on Computer Vision*, pp. 496–501, Corfu (Greece), 1999. 16

[94] M. Pollefeys, F. Verbiest, and L. Van Gool. Surviving dominant planes in uncalibrated structure and motion recovery. In *Proc. ECCV*, pp. 837–851, 2002. 12, 84

[95] M. Pollefeys et al. Image-based 3D recording for archaeological field work. *CG&A*, 23(3):20–27, May/June 2003. 16

[96] L. Quan and T. Kanade. Affine structure from line correspondences with uncalibrated affine cameras. In *IEEE Trans. on PAMI*, vol. 19(8), pp. 834–845, 1997. 5, 12, 112

[97] C. Rother and S. Carlsson. Linear multi view reconstruction and camera recovery. In *Proceedings of the International Conference on Computer Vision*, 2001. 9, 10

[98] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 2000. 68

[99] R. Šára. Accurate natural surface reconstruction from polynocular stereo. In *Proc NATO Adv Res Workshop Confluence of Computer Vision and Computer Graphics*, number 84 in NATO Science Series, pp. 69–86, 2000. 16

[100] R. Šára and R. Bajcsy. Fish-scales: Representing fuzzy manifolds. In *Proc. 6th International Conference on Computer Vision*, pp. 811–817, Bombay, India, January 1998. 16, 29

[101] F. Schaffalitzky and A. Zisserman. Multiview matching for unordered image sets, or, "how do i organize my holiday snaps?". In *ECCV*, 2002. 11, 31, 105

[102] F. Schaffalitzky, A. Zisserman, R. I. Hartley, and P. H. S. Torr. A six point solution for structure and motion. In *ECCV*, vol. I, pp. 632–648, 2000. 24, 39, 53

[103] R. Sedgewick. *Algorithms in C : Part 5 : Graph Algorithms*. Addison-Wesley, Reading, Massachusetts, 3rd edition, 2002. 32, 92

[104] Y. SEO and R. Hartley. A fast method to minimize l-infinity error norm for geometric vision problems. In *Proc. ICCV*, October 2007. 11

[105] K. Sim and R. Hartley. Recovering camera motion using L_∞ minimization. In *CVPR*, vol. 1, pp. 1230–1237, New York , USA, June 2006. 11, 25, 106

[106] K. Sim and R. Hartley. Removing outliers using the L_∞ norm. In *CVPR*, vol. 1, pp. 485–494, New York , USA, June 2006. 11, 25, 105, 106

[107] S. Sinha, M. Pollefeys, and L. McMillan. Camera network calibration from dynami silhouettes. In *Proc. CVPR*, 2004. 74

[108] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3D. *ACM Transactions on Graphics (SIGGRAPH Proceedings)*, 25(3):835–846, 2006. 20, 109

[109] M. Šonka, V. Hlaváč, and R. D. Boyle. *Image Processing, Analysis and Machine Vision*. Thomson, Toronto, Canada, 3 edition, April 2007. 14

[110] H. Stewénius, C. Engels, and D. Nistér. Recent developments on direct relative orientation. *International Journal of Photogrammetry and Remote Sensing*, 60:284–294, June 2006. 20

[111] H. Stewénius, D. Nistér, F. Kahl, and F. Schaffalitzky. A minimal solution for relative pose with unknown focal length. In *CVPR*, vol. 2, pp. 789–794, 2005. 20, 21, 22, 23, 24, 34, 82, 88, 98, 108

[112] H. Stewénius, F. Schaffalitzky, and D. Nistér. How hard is three-view triangulation really? In *IEEE International Conference on Computer Vision (ICCV)*, vol. 1, pp. 686–693, Oct. 2005. 12, 13

[113] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *ECCV96(II)*, pp. 709–720, 1996. 4, 9, 10, 19, 20, 25, 39, 43, 48, 49, 61, 62, 63, 64, 65, 81, 101, 103, 112, 114, 120

[114] R. Szeliski and S. B. Kang. Recovering 3D shape and motion from image streams using non-linear least squares. *Journal of Visual Communication and Image Representation*, 5(1):10–28, 1994. 82

[115] A. W. K. Tang, T. P. Ng, Y. S. Hung, and C. H. Leung. Projective reconstruction from line-correspondences in multiple uncalibrated images. *Pattern Recognition*, 39(5):889–896, 2006. 12

[116] W. K. Tang and Y. S. Hung. A factorization based method for projective reconstruction with minimization of 2-d reprojection error. In *DAGM Symposium*, 2002. 9, 10, 12

[117] W. K. Tang and Y. S. Hung. A subspace method for projective reconstruction from multiple images with missing data. *Image Vision Comput.*, 24(5):515–524, 2006. 9, 10

[118] J.-P. Tardif, A. Bartoli, M. Trudeau, N. Guilbert, and S. Roy. Algorithms for batch matrix factorization with application to structure-from-motion. In *Proc CVPR*, Minneapolis, Minnesota, USA, June 2007. 10, 69

[119] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *IJCV*, 9(2):134–154, November 1992. 3, 9, 10, 44, 96

[120] P. Torr, A. Fitzgibbon, and A. Zisserman. The problem of degeneracy in structure and motion recovery from uncalibrated images. *IJCV*, 2000. 12, 84

[121] P. H. S. Torr. Model selection for two view geometry: A review. In *Shape, Contour and Grouping in Computer Vision*, pp. 277–301, 1999. 12, 84

[122] P. H. S. Torr and A. W. Fitzgibbon. Invariant fitting of two view geometry. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5):648–650, 2004. 9

[123] P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78:138–156, 2000. 36

[124] B. Triggs. The geometry of projective reconstruction I: Matching constraints and the joint image. Submitted to IJCV, 1995. 112

[125] B. Triggs. Factorization methods for projective structure and motion. In *CVPR*, pp. 845–851, 1996. 5, 12, 111, 112

[126] M. Urban, T. Pajdla, and V. Hlaváč. Projective reconstruction from n views having one view in common. In *Vision Algorithms: Theory & Practice*, vol. 1883 of *LNCS*, pp. 116–131, Corfu, Greece, September 1999. 112

[127] M. Urban, T. Pajdla, and V. Hlaváč. Projective reconstruction from multiple views. In *Technical report CTU-CMP-1999-5*, December 1999. 4, 9, 10, 43

[128] M. Uyttendaele, A. Criminisi, S. B. Kang, S. A. J. Winder, R. Hartley, and R. Szeliski. High-quality image-based interactive exploration of real-world environments. *CG&A*, 24(3):52–63, May/June 2004. 11, 87, 98

[129] J. Šivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV03*, pp. 1470–1477, 2003. 31

[130] T. Werner. Constraint on five points in two images. In *CVPR 2003: Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. II, pp. 203–208, Madison, USA, June 2003. 23

[131] T. Werner and T. Pajdla. Cheirality in epipolar geometry. In *Proceedings of International Conference on Computer Vision*, pp. 548–553, Vancouver, Canada, July 2001. 35, 76, 80

[132] T. Werner and T. Pajdla. Oriented matching constraints. In *BMVC*, pp. 441–450, Manchester, UK, September 2001. 48, 63, 76, 80, 90

[133] R. Yang and M. Pollefeys. Multi-resolution real-time stereo on commodity graphics hardware. In *Proc. CVPR*, vol. 1, pp. 211–217, June 2003. 17

[134] C. Zach, T. Pock, and H. Bischof. A globally optimal algorithm for robust tv-l1 range image integration. In *Proc. ICCV*, October 2007. 17

[135] W. Zhang and J. Kosecka. Generalized ransac framework for relaxed correspondence problems. In *3DPVT '06: Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, pp. 854–860, 2006. 23

[136] H. Zhong and Y. Hung. Factorization-based hierarchical reconstruction for circular motion. In *BMVC 2004*, 2004. 12